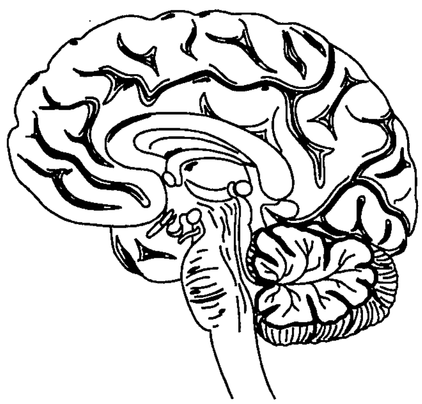
COMPUTER SCIENCE

USING MACHINE LEARNING TO DETERMINE UNITED STATES ARMY READINESS AT THE BATTALION LEVEL



CPT Todd L. Smith

Paper under the direction of Professor Doug Fisher

Approved _Doug Fisher_ Date _Oct. 18, 1996_

USING MACHINE LEARNING TO DETERMINE

UNITED STATES ARMY READINESS

AT THE BATTALION

LEVEL

By

CPT Todd L. Smith

Paper

Submitted to the Faculty of the

Graduate School of

Vanderbilt University

in partial fulfillment of the requirements

for the degree of

Master of Science

in

Computer Science

December, 1996

Nashville, Tennessee

Approved:

Date: Oct. 18, 1996

Acknowledgments

This research project was completed with the assistance of many individuals. Special thanks to Dr. Doug Fisher for his guidance and advice, to Lieutenant Colonel Michael Patenaude for providing the time and expertise to rate the sample units, and to MAJ Richard Daum, CPT Tony Potts, CPT Bob Doerer, and CPT V.J. Tedesco for their help with the survey. I would also like to thank all the professors in the Computer Science Department for their daily commitment to excellence in teaching and research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter I

Background

This paper develops a model for determing the combat readiness of an Army Battalion and investigates the problem of automating this task. The central problem is one of classification and the domain is that of military readiness in United States Army Battalions, but it could be applied to a wide variety of problems with only minor modifications. The problem is simply defined but very difficult to solve.

It is first necessary to understand the current method for evaluating unit readiness and the problems associated with it. Readiness reports are called Unit Status Reports (USRs) in the Army, and are conducted at the Battalion level. Each service has a different report with its own problems. I will address only the Army's method.

Battalions are the lowest level of organization in the Army that is considered able to support its own operations and is probably the highest level where commanders are fully aware of the condition of the unit. It consists of three to eight hundred soldiers with equipment dependent on its mission, i.e., Aviation, Armor, Infantry, or Administration. Table 1-1 compares a few typical battalions.

	Approx # of Soldiers	Major Piece of Equipment	Number of Major Equip
Aviation	~300	Helicopter	30
Armor	~400	Tank	58
Infantry	~500	Infantry Fighting Vehic.	54
Light Infantry	~500	Light Truck	60
Admin	~800	Light Truck	Varies

Table 1-1: Typical Battalions

As Table 1-1 shows, battalions are highly unique depending on their mission. What is important for an Administration Battalion might not be important for an Armor Battalion. Commands located above Battalion level are not pure, i.e., they may contain different types of units. This makes it difficult to report a consistent readiness above battalion level. The purity of battalions and their potential for independent operations make them an obvious level for status reports.

Each month the battalion's key personnel gather information from a variety of sources, analyze it in excruciating detail, and determine the combat readiness of the unit on a scale of 1 to 5. The report includes information through the 15th of each month, but the process requires that units begin preparation during the first week of each month and finish the report prior to the 11th or 12th. The additional information is extrapolated. Information is examined in three broad areas: Logistics, Training, and Personnel.

Logistical information includes the equipment on hand compared to the equipment that is authorized to the unit, the serviceability of that equipment (maintenance rates), and evaluations of the units expendable supplies. Certain pieces of equipment are known as Pacing Items and must maintain a certain on-hand

percentage and readiness rate. A tank unit has to have a certain number of tanks. A battalion can have several hundred types of equipment and several hundred pieces of equipment per type. Many of the pieces require complex maintenance reports that must be compiled for the monthly report. The equipment information is located in a consolidated database in either MS DOS or Burroughs Twenty Operating System format. A single database can contain the property data for up to thirty battalions. The Battalion motor pool manages the maintenance data in a variety of formats. Primarily, they have MS-DOS based personal computers using the Unit Level Logistics System database.

Personnel information includes the number of soldiers available per specialty and rank. Very few units have one hundred percent of their authorized personnel and some have less than eighty percent. If the shortages are in key areas or concentrated in one area, the results can be severe. Also, soldiers have requirements that take them away from the unit or render them nondeployable. These factors must be compiled and a status assessed. The data is maintained in the personnel or S1 office in both MS-DOS format on personal computers and on a tactical system using the Burroughs Twenty Operating System (BTOS).

Training is the most subjective of the areas, and unit leaders must examine a wealth of training information prior to judging the effectiveness of the unit. This information includes weapons qualifications, physical fitness reports, unit training results, and unit specific tasks such as pilot training. These records are normally stored in a MS-DOS based format on personal computers located in the training or S3

office or in the underlying units. Figure 1-1 is a simplified Data Flow Diagram for the battalion's 2715 input.

After the battalion collects the information, the staff officers compile it into an understandable report, and the key leaders evaluate their areas of responsibility, the battalion commander makes a decision on the readiness of his unit. His decision is part objective and part subjective, based on his interpretation of the information and his intuition. This report is then reviewed by the controlling organizations before being forwarded to the Department of the Army.

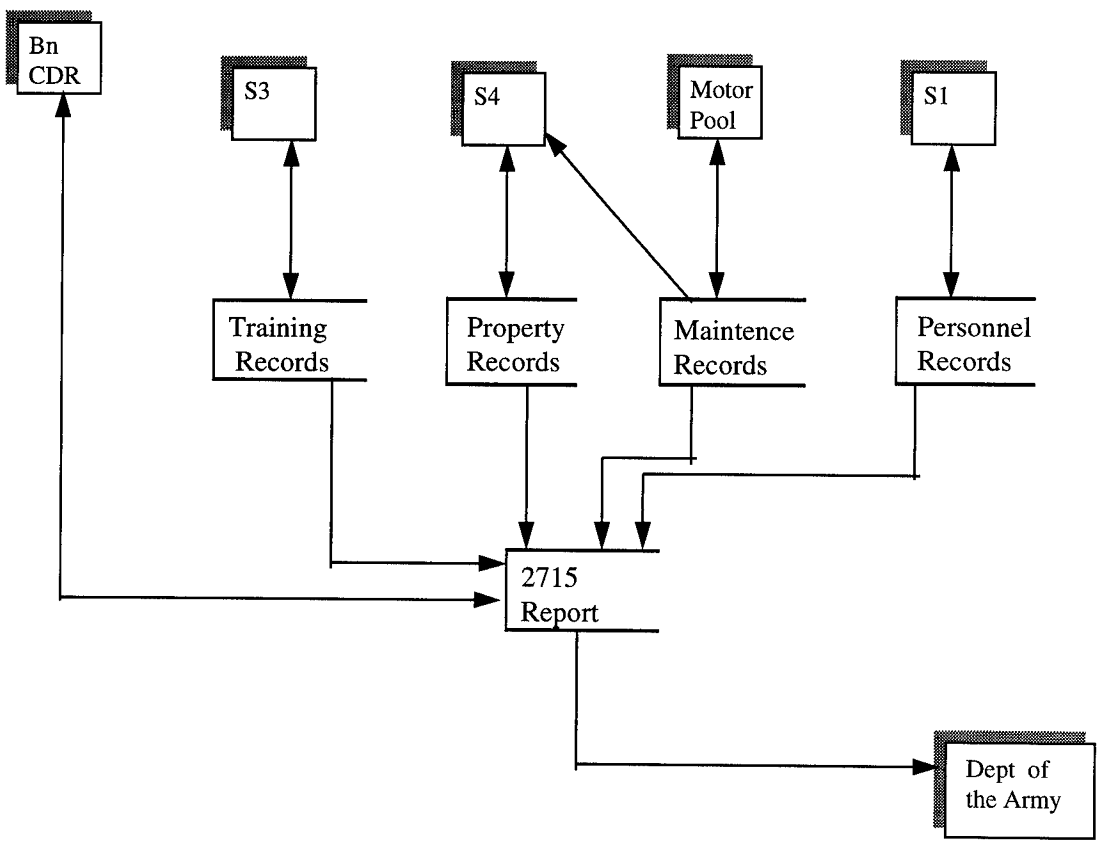


Figure 1-1 - Data Flow Diagram for a Battalion's 2715 reporting

After the report is forwarded to HQDA, they use it in a variety of ways. [AR 220-1, 93] defines the objectives of the Unit Status Report. These include:

(1) Indicate the Army-wide conditions and trends.

(2) Identify factors which degrade unit status.

(3) Identify the difference between current personnel and equipment assets n units and full wartime requirements.

(4) Assist HQDA and intermediate commands to allocate resources.

Often, HQDA and Congress use these reports to determine funding issues and as justification for expenditures.

What are the problems associated with the report? It seems straight forward for a unit to report its condition, forward that report to Washington, and have their report compared to other Army units for resource allocation.

The first major problem is determining what readiness means. [Betts, 95] observes that "readiness is easiest to assess at the lowest level: individual soldiers" and it grows more difficult with the size of the unit. A soldiers has basic tasks with known standards. If he can accomplish each task to the standard, then he is considered trained. A weapon system, such as an Attack Helicopter, is not as simple. It has a crew that must be evaluated, a maintenance condition, and a support structure that must keep it in operation. To what degree must the helicopter be combat ready? [Betts, 95] states that the "technological sophistication of many modern weapon systems means that very few are ever likely to be fully mission capable." So even at the weapon system level, readiness requires judgment.

Now consider the battalion level with different types of soldiers, weapon systems, and missions. [Holz, 94] describes three problem areas when evaluating a unit:

(1) Multitudes of missions and tasks.

(2) Uniqueness of each unit and its post.

(3) Difficulty of measuring leadership and cohesion.

The answer has been to require all the available information from units and have them perform a subjective evaluation of the information within defined parameters. [Betts, 95] points out that "the volume and redundancy of the reporting requirements have sometimes been so great that they overburden commanders and reduce productivity." This is the second problem with the current report: it places a burden on the units.

Third, there is a problem with shifting standards. In the 1980s, the Army converted from M1 tanks to M60 tanks. Readiness went down in each of the new units because units were transitioning to new types of equipment. The units might be required to have a different type or number of communication systems, for example. The regualations required that the units report a reduced level of readiness even though the M1 was vastly superior to the M60. Combat potential increased but readiness reports showed a marked decrease. The Army fields new equipment each year, and the readiness baseline moves with each fielding. It is difficult to measure progress with shifting standards.

A related problem is the subjective nature of the standards. Commander's perceptions of readiness may differ as does their personal optimism. If the data indicates a unit is C2, one commander may downgrade the unit to C3 because he "feels" the unit has major deficiencies. Another commander may upgrade the same unit to C1 because he feels it has cohesion that will overcome its problems.

Fourth, careerism tends to alter the results. [Betts, 95] states that self interest can move the ratings either direction. "New commanders would have an interest in a poor rating" so that they can show improvements later. "Commanders near the end of their tour would have an interest in a high rating" to demonstrate their managerial success. [Betts, 95] believes the "dominant tendency seems to be to inflate ratings."

Finally, "there is no agreement on what indices provide the best measure of operational readiness" [Betts, 95]. In a report to Congress, the General Accounting Office recommended that the Secretary of Defense develop a more comprehensive readiness measurement system. This report included several indices that should be included, and directed the Defense Department to study additional indices. This work is ongoing [NSIAD, 95-29].

Chapter II

Unit Status Report Survey

Purpose

I conducted this survey to determine the opinions of the officers who prepare the Unit Status Reports (USR) each month. In ten years of service, I have observed a common criticism that the USR requires a disproportionate amount of time compared to the results that filter back down to the unit. It was my intent to quantify this opinion.

Collection of Data

The collected data does not represent a random sample, and I have not included a margin of error. The internet was the primary source of information including both direct mailings to the Command and General Staff College at Fort Leavenworth, Kansas, and the United States Military Academy at West Point, and a posting to the Army Automation List Server. Approximately 150 surveys were sent to the academic institutions, and 110 officers responded. However, some of the responses indicated that the survey was not applicable. I used Email listings from the institution's World Wide Web page, so some of the addresses could have been out of date. The Automation List Server has over 500 subscribers, but it is impossible to say how many follow the daily postings. 60 officers responded. Additionally, surveys were distributed to individuals at Fort Campbell (15), Vanderbilt University's ROTC

department (8), and the University of Kentucky's ROTC department (5). A total of 164 surveys were collected from these sources.

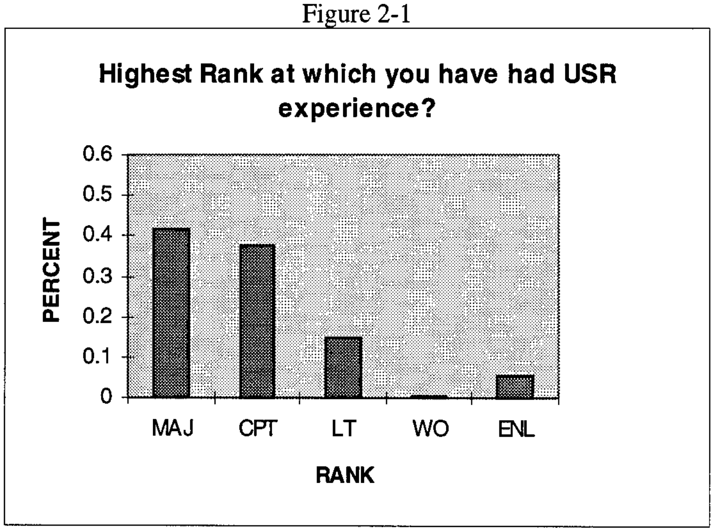
Results

1. What is the highest rank at which you have had USR experience?

Table 2-1

	MAJ	CPT	LT	CWO	Enlisted
Percentage	41 %	38 %	15 %	1 %	5 %
Number	68	62	24	1	9

Figure 2-1



The majority of surveys that were returned with "not applicable" were from enlisted service members and warrant officers. Only two lieutenants indicated that they had never had any USR experience. There were no officers above the rank of Lieutenant that had not had USR experience.

There were several officers that had USR experience at the Lieutenant Colonel level. These were included in the Major totals. Seventy-nine percent of the surveys were completed by officers having a rank of Captain or greater. There are two

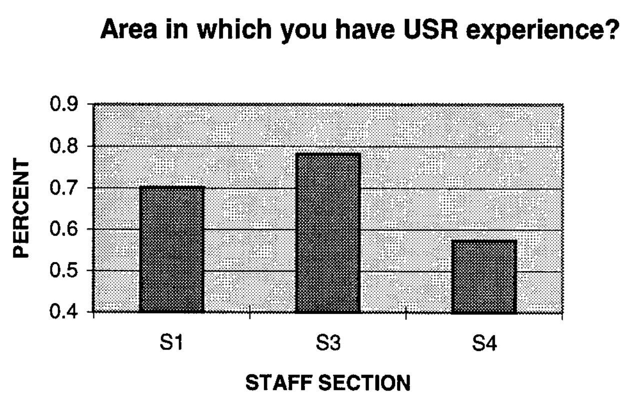
reasons for this discrepancy, (1) Staff officers and commanders are usually responsible for the preparation of the USR, and most staff officers and commanders have these ranks. (2) The majority of the individuals contacted for the survey were commissioned officers.

2. In what area do you have USR experience?

Table 2-2:

	Personnel	**Training**	**Logistics**
Percentage	70 %	78 %	57 %
Number	115	128	94

Figure 2-2:



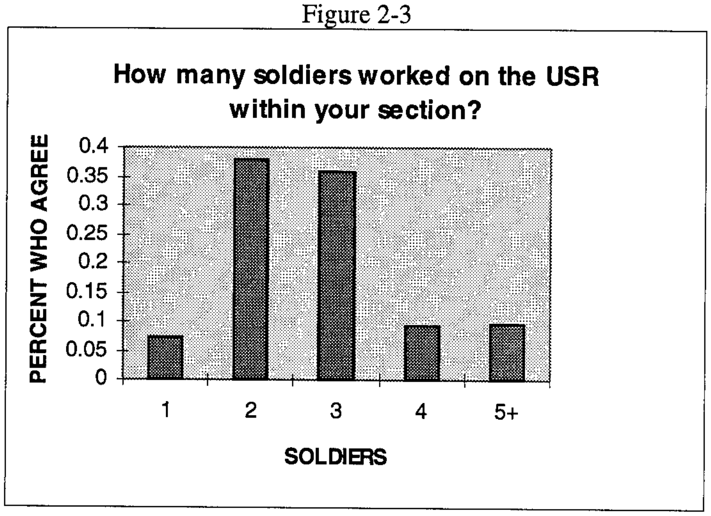
Officers were allowed to check all areas in which they had experience. The majority of individuals had experience in all three of the subject areas, but training was the most common choice, followed by personnel and logistics. The S3 section typically has the most officers and assumes overall responsibility for the USR.

3. How many soldiers worked on the USR within your section (i.e., Personnel, Training, or Logistics)?

Table 2-3

	1	2	3	4	5+
Percentage	7 %	38 %	36 %	9 %	10 %
Number	12	62	59	15	16

Figure 2-3



The results formed a normal distribution with a mean response of 2.8 soldiers per section required to complete the USR and a standard deviation of 1.05. This indicates that for the three sections, an average of 8.4 soldiers per month were required together with the executive officer to oversee and coordinate the activities of the staff.

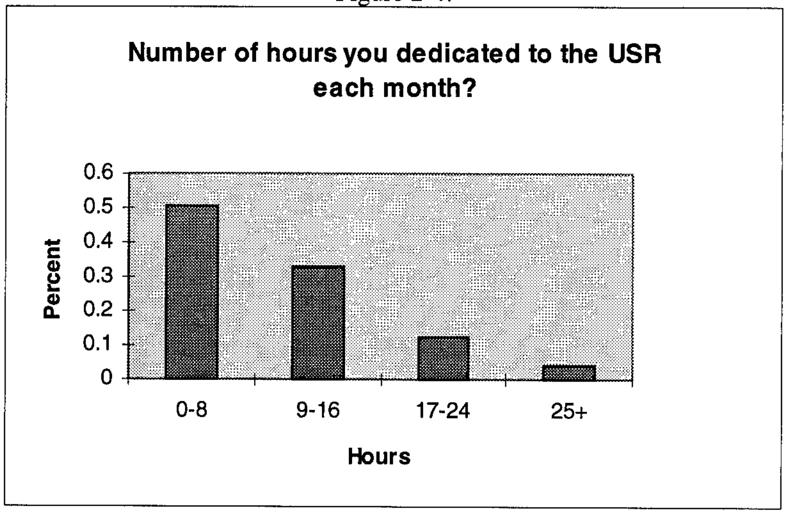
4. How many hours did you dedicate to the USR each month?

Table 2-4:

Hours:	0-8	9-16	17-24	25+
Percentage:	50 %	32 %	12 %	1 %

Number:	83	54	20	7

Figure 2-4:



Using the midpoint of each range (4 for 0-8, 12 for 9-16), the mean number of hours reported was 9.5 per month per individual.

5. To what extent did the Unit Status Report return tangible results to your unit?

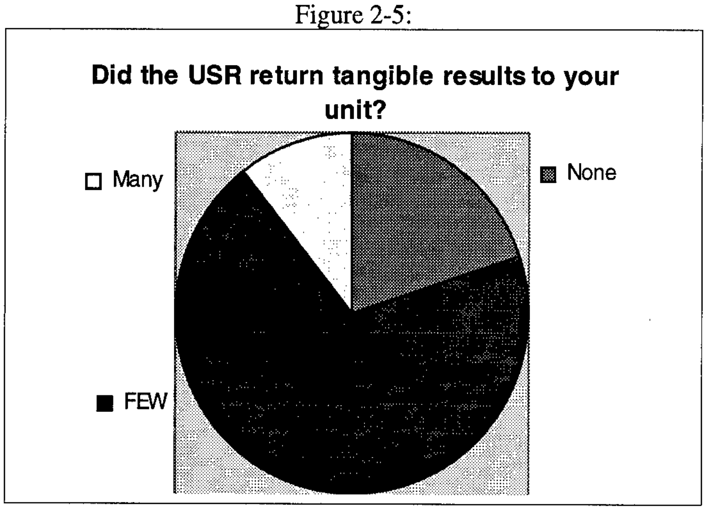
Eighty-nine percent of responders indicated that the Unit Status Report returned either Few or No positive benefits. Of those that indicated Few or Many, many stated that the process of reviewing the data was the primary benefit. Only a small percentage claimed that headquarters above division level ever responded to a problem. However, the USR is not primarily a tool for assisting units. It is intended to provide readiness information to Headquarters, Department of the Army (HQDA). AR 220-1 clearly establishes this in its objectives. The last objective listed is to

"assist HQDA and intermediate commands to allocate resources" [AR 220-1, 93]. The remaining objectives concern providing information about Army units.

Table 2-5:

	None	Few	Many
Percentage:	20 %	69%	11 %
Number:	32	111	17

Figure 2-5:



6. *Was the USR a training distracter? Selections were "Yes, the time could have been better utilized" or "No, it provided insights into the readiness of the unit."*

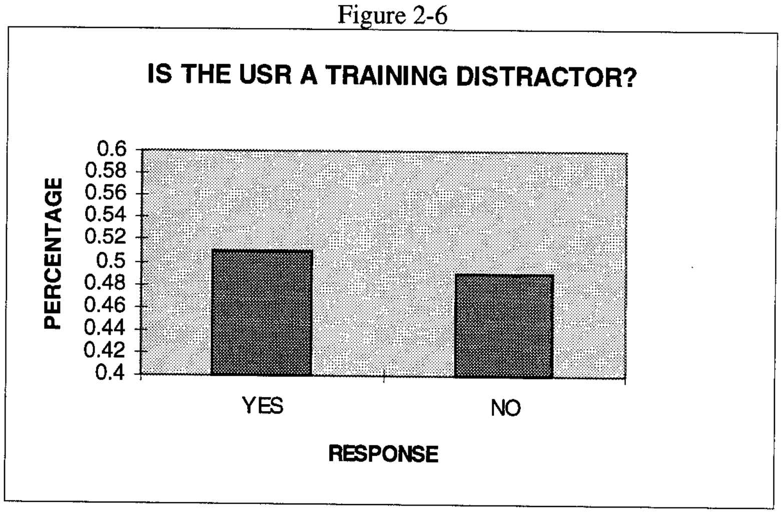
Responders were divided on this question. A few individuals pointed out that the questions was misleading because the USR provided both insights and distractions. My belief is that most selected the response they thought most relevant. Surprisingly, Majors selected Yes fifty-five percent of the time compared to the total mean of fifty-one percent. One could make an argument that experience in the Army would increase the understanding of USR requirements. As one moves up and

performs a variety of jobs, she would gain insights not seen by lower ranks. However, this data suggests that there is no statistical significance between rank and attitude.

Table 2-6:

	Yes	No
Percent	51 %	49 %
Number:	81	78

Figure 2-6



7. Was USR data ever inflated in your unit?

There are several reasons why data might not be accurately reported. A unit might overly emphasize a problem area in order to give it visibility and possibly speed a solution. This would be most probable in the personnel and equipment on-hand areas where higher level intervention is required for a solution. Another possibility is a unit failing to examine data until it is time to report it, and feeling that the data does not reflect the readiness of the unit. This scenario is more likely in the training area.

For example, units routinely spend several consecutive months in the field. It is difficult during this time to ensure that 90% of the soldiers take a physical fitness test; however, commanders are reluctant to report a low readiness rate because of physical fitness. To correct the deficiency, commanders can either falsify the report or have their soldiers take a fitness test with little warning. Of the three possible actions, (1) report the truth and face the consequences, (2) falsify the report, or (3) make the soldiers take a short notice fitness test, option 2 is sometimes selected.

Equipment Readiness is a potential area for erroneous reporting as well. There is a huge reporting requirement and the standards, although clear, are often debatable. A unit might not report a piece of equipment as Not Mission Capable when the regulation states that it is.

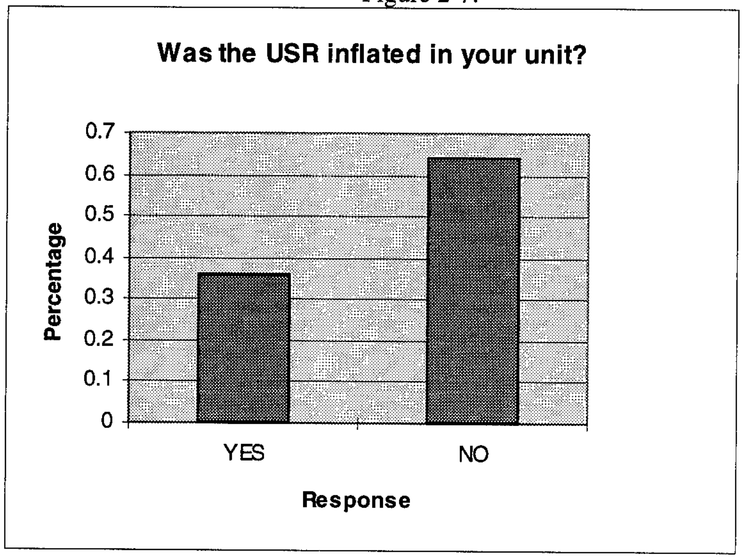
[Betts, 95] states that, "The difficulty associated with aggregating measurements in general, as well as the career incentives that those who gather data have to fabricate or distort, should make people skeptical about what ostensible information about readiness really shows." There exist certain truths: (1) Units make data collection errors, (2) Some data is over-emphasized so that it receives attention, and (3) Some data is omitted to protect careers. These truths tend to reduce the value of the readiness reports.

In this poll, forty percent of Majors and Captains claimed that reports were inflated while only thirty percent of the Lieutenants made the same claim. It may be that the more experienced officers were more involved with decisions of this type.

Table 2-7:

	Yes	No
Percentage:	36 %	64 %
Number:	57	101

Figure 2-7:



Thirty-six percent of responders indicated that the USR was inflated. It is possible that this number would have been higher had it been stressed that inflated meant up or down. Also, this does not include the errors that are made.

8. Could an automated system that looks only at data evaluate unit readiness?

Seventy-four percent of responders indicated that an automated system could not evaluate unit readiness. This shows the common belief that a commander "knows" his unit and gets a "feel" for it that numbers cannot measure. This "feel" not only includes certain intangibles such as morale, espirit, and teamwork, but it also includes an overall sense of mission accomplishment. For example, the unit might

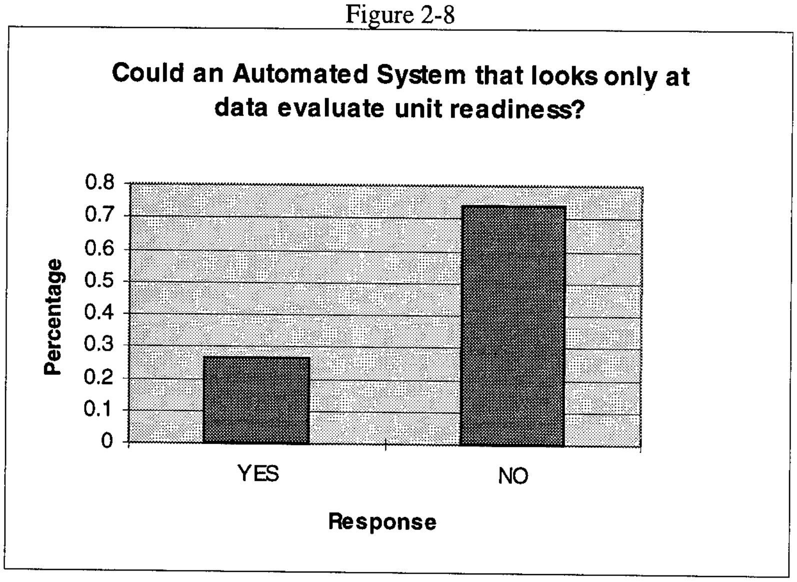
have had a serviceability percentage that was seventy-five percent, but the commander thinks his motor pool is superb and would perform exceptionally in a wartime environment. No one argues with the collected statistics, but many claim that they do not capture the warfighting potential of the unit.

Most surprisingly, sixty-three percent of those that said the USR was inflated and distracting also said that an automated system could not evaluate unit readiness. This shows how firmly entrenched the perception is that a commander must evaluate his unit. Even those that admit the current system is imperfect believe it is better than an automated system.

Table 2-8:

	Yes	No
Percent:	26 %	74 %
Number:	41	114

Figure 2-8



Summary

1. The poll suggested that, on average, 9.4 personnel utilize 9.5 hours each month preparing the USR for a total of 89.3 man-hours.

2. Eighty-nine percent reported that the USR returned Few or No positive benefits to the unit.

3. Fifty-one percent reported that the USR was a training distracter.

4. Thirty-six percent reported that the USR had been inflated in their units.

5. Seventy-four percent said that an automated system that looked only at readiness could not replace the current system. Surprisingly, of those that said the USR was both a distracter and inflated, sixty-three percent still said that an automated system could not evaluate unit readiness.

Chapter III

Criteria for Unit Assessment

Overall Unit Ratings

There are five categories of readiness, C1, C2, C3, C4, and C5. C5 indicates a unit that is in a reorganization status either because it is disbanding or being created. It is not used by normal field units. The definitions of the categories are as follows:

C1: The unit is fully combat ready.

C2: The unit is combat ready but has minor deficiencies.

C3: The unit has major deficiencies.

C4: The unit is not ready for combat.

Army Regulation 220-1 establishes the criteria for evaluating units. There are several rules that dictate levels, but there are many subjective areas. Additionally, commanders can increase or decrease the overall level if he believes it is justifiable.

The regulation groups the performance indicators into three areas: personnel, training, and logistics. Each area receives a rating that is equal to the lowest of its indicators, and the overall rating is equal to the lowest rating of the areas. The battalion commander is responsible for evaluating the data and determining the true readiness of the unit. The following sections describe the three areas.

For the experiments described in this paper, an Army Lieutenant Colonel evaluated 120 data sets each comprising 38 indicators. He used the following descriptions, AR 220-1, and eighteen years of experience including a battalion

command to evaluate the data sets. Appendix A describes the data sets in greater detail.

Personnel

The purpose of the personnel section is to determine the status of the unit's personnel. A high rating indicates that the unit has enough soldiers, the soldiers have the right specialties and ranks, and the transition rate is sufficiently small. Each paragraph concludes with a description of the data as either continuous or discrete, and the range of the variable.

1. *Assigned Strength Percentage*: The unit's assigned strength divided by the required strength. The required strength is based on the unit's MTOE (Modified Table of Organization and Equipment.) Continuous. Range: 0-100.

Standards: C1 - 100-90%

C2 - 89-80%

C3 - 79-70%

C4 - 69% and below.

2. *Available Strength Percentage*: Available strength divided by the required strength. The number available is the number of assigned personnel minus those that are not able to deploy in support of the unit's wartime mission. Continuous. Range: 0-100.

Standards: C1 - 100-90%

C2 - 89-80%

C3 - 79-70%

C4 - 69% and below

3. *MOS Qualified Percentage*: Qualified strength divided by the required strength. A soldier is not qualified if his MOS or his Additional Skill Identifier is not appropriate for his assignment. Soldiers can fill spaces that are two above and one below his/her current grade. Continuous. Range: 0-100.

Standards: C1 - 100-85%

C2 - 84-75%

C3 - 74-65%

C4 - 64% and below

4. *Available Senior Grade Percentage*: The number of assigned commissioned, warrant, and noncommissioned officers divided by the required number. Continuous. Range: 0-100.

Standards: C1 - 100-85%

C2 - 84-75%

C3 - 74-65%

C4 - 64% and below

5. *Personnel Turnover Rate:* This indicator measures the turmoil caused by transitioning personnel. It is equal to the number of personnel that have joined the unit in the 3 months prior to the "as of" date divided by the assigned strength. Continuous. Range: 0-100.

Standard: This indicator is subjective, but less than 10% is optimal.

Logistics

The purpose of the logistics section is to determine the availability and condition of the unit's equipment. The type and importance of the unit's equipment depends on the unit's mission. The rating consists of a supply area (S) and a maintenance area (R).

1. *Equipment On Hand (EOH):*

(a). Each type of equipment is assigned an Equipment Readiness Code (ERC) of P, A, B, or C. For each ERC P and A equipment type, divide the quantity on hand by the required quantity. Assign each line a rating based on the percentage fill:

S1 - 100-90%

S2 - 89-80%

S3 - 79-70%

S4 - 69-60%

Count the number of lines for each category. Divide the number of S1, S2, S3, and S4 lines by the total number of lines. Continuous. Range: 0-100.

Standards: S1 - The number of S1 lines divided by the total number of lines is greater than or equal to 90%

S2 - The number of S1 lines is less than 90% of the total, but the number of S1 plus S2 lines divided by the total lines is greater than 85%.

S3 - The number of S1 plus S2 lines is less than 85% of the total, but the S1 plus S2 plus S3 lines divided by

the total lines is greater than 80%.

S4 - The number of S4 lines is greater than 20% of the total, or conditions for S3 are not met.

(b). Pacing Items are computed separately. The EOH rating cannot be higher than the lowest S rating of the Pacing Items. A Pacing Item is identified by an ERC of P on the unit's MTOE. The data sets will include three pacing items per unit. Continuous. Range: 0-100.

Standards: S1 - 100-90%

S2 - 89-80%

S3 - 79-70%

S4 - 69% and below

2. Nuclear, Chemical, and Biological Equipment (NBC): The S level for six items of NBC equipment: Mask, Detector, Decontamination, Protective Suit, Medical, and Radiac.

Standards: S1 - 100-90%

S2 - 89-80%

S3 - 79-70%

S4 - 69% and below

3. Equipment Serviceability: The total number of available hours (days) divided by the total number of possible hours (days) for reportable equipment. This measures the readiness of the unit's equipment. The equipment serviceability of aircraft is measured separately. Continuous. Range: 0-100.

Standards:	R1 - 100-90%	Other than Aircraft
	R2 - 89-70%	
	R3 - 69-60%	
	R4 - 59% and below	
Standards:	R1 - 100-75%	Aircraft
	R2 - 74-60%	
	R3 - 59-50%	
	R4 - 49% and below	

4. Prescribed Loads List (PLL): The number of types of PLL that are zero balance (none on hand) divided by the total number of types. There is a separate indicator for ground and air items. Continuous. Range: 0-100.

Standards: Subjective evaluation with goal of less than 10% zero balance.

Training

The purpose of the training section is to determine the training status of the unit, i.e.,, its ability to accomplish the tasks identified in the unit's Mission Essential Tasks List (METL).

1. METL evaluation: The unit has a minimum set of required tasks that they must be able to accomplish to fulfill their wartime mission. The unit assigns a grade of Trained, Partially Trained, or Untrained to these tasks. The number of METL tasks is determined by the unit, but the data sets will include three tasks for each unit. Discrete. Range: Trained, Partially, Untrained.

Standards: Subjective. C1 units should not have any Untrained METL Tasks and very few Partially Trained METL tasks.

2. *Physical Fitness Scores.* Each soldier takes a PT test with a maximum score of 300 points. A soldier qualifies if he scores over 180 points. The following standards are for qualification. Continuous. Range 0-100, and 0-300.

Standards: T1 - 100-90%

T2 - 89-80%

T3 - 79-70%

T4 - 69% and below

Additionally, the assessors evaluate the average PT Score for a unit. 250 is a goal for unit average.

3. *Basic Rifle Marksmanship:* A soldier is qualified if he/she receives a score of Expert, Sharpshooter, or Marksman. Standards represent percent qualified personnel. Continuous. Range: 0-100.

Standards: T1 - 100-90%

T2 - 89-80%

T3 - 79-70%

T4 - 69% and below

Percentage of Expert, Sharpshooter, and Marksman are subjective.

4. *Aviator Readiness:* Each aviator has a Readiness Level of RL1, RL2, or RL3 for both his mission tasks and Night Vision Device tasks. Goals depend on the particular unit's mission.

Continuous.

Range: 0-100.

Standards:	T1 - 100-90%
Mission Tasks	T2 - 89-80%
	T3 - 79-70%
	T4 - 69% and below
Standards(NVD):	Depends on unit mission, but normally 30%

5. Training Events: Measures the frequency of key training events. Listed as months since last (1) ARTEP, (2) NTC, CMTC, or Division Training Exercise, (3) Other Field Training Exercise, (4) Gunnery, and (5) Command Training Exercise. Discrete. Range: 0, 1, 2, 3 ,..., 24.

Standard: Subjective.

6. Crew Weapons: The number of trained crews divided by the number of required weapon systems. If there are more trained crews than weapon systems, then the value is recorded as 100 percent. The data sets will include three crew weapons. Continuous. Range: 0-100.

Standards:	T1 - 100-90%
	T2 - 89-80%
	T3 - 79-70%
	T4 - 69% and below

7. Leadership Training: Measure of the impact that the availability of qualified leaders is having on the unit. Discrete. Range: A, B, C, D

Standards: Subjective.

A - Insignificant Impact

B - Minor Impact

C - Major Impact

D - Unable to meet the METL requirements because of a lack of qualified leaders

Summary

There are a total of thirty-eight indicators in this model. For the data used in this paper, personnel has five indicators. Logistics has fifteen indicators in six areas. Training and Readiness has eighteen indicators in 10 areas. The regulation stipulates that the overall rating can be no higher than the lowest of the three areas, and the individual area rating can be no higher than the lowest category. [Betts, 95] observes that since the "composite must equal the lowest of the individual ratings, it tends to understate readiness." The commander does have the flexibility to move up or down one rating (overall) to compensate for this tendency.

Table 3-1: Summary of Criteria

Criteria	C1	C2	C3	C4	Hard/Soft
ASPER	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
AVPER	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
MOS %	**100-85%**	**84-75%**	**74-65%**	**Below 64%**	**Soft**
SG %	**100-85%**	**84-75%**	**74-65%**	**Below 64%**	**Soft**
Turnover					**Soft**
EOH	**100-90%**	**89-85%**	**84-80%**	**C4 > 20%**	**See Chart**
PI-EOH	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
NBC EOH	**100-90%**	**89-85%**	**84-80%**	**C4 > 20%**	**Soft**
ES	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
PI-ES	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
PLL-0-BAL	**<10%**	**<20%**	**<30%**	**>30%**	**Soft**
METL Eval					**Soft**
PT-Qual'd	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
PT-AVG	**>250**	**>230**	**> 210**	**< 210**	**Soft**
BRM-Qual'd	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
BRM % Exp					**Soft**
AVN RL1 %	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Hard**
AVN NVG%	**> 30%**				**Soft**
Months Since Event				**> 18?**	**Soft**
Crew Weapons	**100-90%**	**89-80%**	**79-70%**	**Below 69%**	**Soft**
Leadership Training	**A**	**B**	**C**	**D**	**Soft**

Chapter IV

Problem Statement

Problem

There exists a need to automate the collection and analysis of the information in the battalion and provide an intelligent judgment as to the status of the individual areas and the overall readiness of the unit. The advantages includes a common yardstick for all Army units rather than subjective evaluations, a reduction in the time required of the key leadership in producing the report, and a more comprehensive examination of the physical data. Only recently has the level of automation in the lower echelons of the Army become sufficient to realize this objective.

If the standards are fully described in AR 220-1, why is it difficult to apply the standards to an Army unit? The difficulty is that the standards do not include the experience and judgment of the commander and staff. The commander does not look at a single category and deduce a rating. Rather, he uses a parallel approach to examining the features. Also, units have different missions and different requirements for personnel and equipment. The regulation can not cover every variation, so it is general enough to be used by all units. Finally, many standards are not fully specified. Mission Essential Task evaluation is probably the most important indicator of unit capability, but the evaluation standards are completely subjective. The evaluator for the data sets used in this model deviated from the regulation on

26% of the examples, and this was a controlled environment without the usual prejudices that accompany battalion command.

In summary, we can classify the above problem as inaccessible, nondeterministic, episodic, highly dynamic, and continuous. The inputs to the system are constantly changing and soldiers continuously update their respective databases. Capturing a snapshot is a problem with the current system that an automated system could standardize. The report is episodic in that it is a monthly requirement, and last month's results do not impact upon the current month. An intelligent system should check for consistencies, however. Nondeterminism stems from the inability of the system to predict future actions based on current data. A maintenance status that is great today could fall catastrophically tomorrow. Inaccessibility occurs because clerks either fail to update the databases or are not timely thus insuring that the system will never have a complete picture of the actual unit. Problems displaying these characteristics are very difficult for automated processes to solve, and many commanders believe that their intuition is a key factor in the evaluation process.

Experimental Design

For the experiments, I used generated data as described in Appendix A. I used the generated data rather than actual unit data because

(1) Unit data would have had a confidential classification,

(2) Units don't typically keep a database of the indicators, only the report itself,

(3) It would be very difficult to obtain a balanced distribution of samples.

A previous battalion commander evaluated the data and assigned a classification based on the standards and his experience. The goal of the learning algorithms was to learn not only the rules but the experience.

Each experiment used a stratified, cross validation technique dividing the data into six sections of twenty vectors. The experiment was repeated six times. Each time a different section comprised the test data while the remaining five sections comprised the training data.

Overview of Systems

I evaluated four learning algorithms: (1) Neural Networks using the backpropagation algorithm, (2) Decision Tree Induction, (3) Bayes's Classification, and (4) Classification based on the nearest neighbor concept. Additionally, I experimented with a binary classification scheme that evaluated the difficulty of determining whether a data set belonged to a particular class. For comparison, I also used a rule based classifier consisting of the "hard" standards described earlier. The rule based classifier correctly classified 74% of the data sets.

Chapter V

Implementation Using Neural Nets

Introduction to Neural Networks

Appendix B provides a history and description of neural networks. They are the subject of much current research because they exhibit fault tolerance, have a highly parallel approach to problem solving, are adaptive, and can handle contextual information [Haykin, 1994]. A neural network can be given a set of input, output pairs and learn the relationship between the pairs even when the relationship is not known. They have proven ideal for feature detection because different cells in the network can be trained to identify features of a pattern. In this respect, neural networks seem ideal for this problem.

Results

Data Representation

The raw data consisted of alpha-numeric characters that were not useful in the neural network context. Neural Networks can accept and output either binary or continuous (scaled between 0 and 1) numeric data, so representation of the data is obviously highly important. I experimented with both continuous and binary schemes.

Continuous Inputs

My first attempt was to scale the inputs to a range of 0 to 1 for input into the neural net. The output of the system was binary with 1-0-0-0 mapping to C1, 0-1-0-0 mapping to C2, 0-0-1-0 to C3, and 0-0-0-1 to C4. A sample input is 89% which was presented to the system as .89.

After experimenting with several different network configurations, it became apparent that the net would not converge using continuous inputs. The error rate was consistently above 80% after six hours training. Annealing did not improve the convergence characteristics. Probably, the resolution of the inputs was such that continuous inputs were inappropriate. For example, an assigned personnel percentage of 90% is C1 while 89% is C2. This degree of resolution was impractical. While scaling the values within the certain ranges might improve performance, it would introduce an unacceptable bias to the data which might obscure judgments made by the expert. For example, if the expert decided that, based on the other data, an 89% data item was sufficiently close to 90% to achieve a C1 classification, the scaling would hide this criteria.

Preprocessor

Similar to [Gorman, 1988], I coded a preprocessor to convert the 38 input representation into a 149 input representation using domain knowledge. The processed information was binary with four bits representing each continuous input (three inputs required only three bits). I based the conversions on the individual standards given by Army regulations, technical manuals, field manuals, or standard

procedures for the specific categories. Figure 5 -1 is a graphical view of the problem structure.

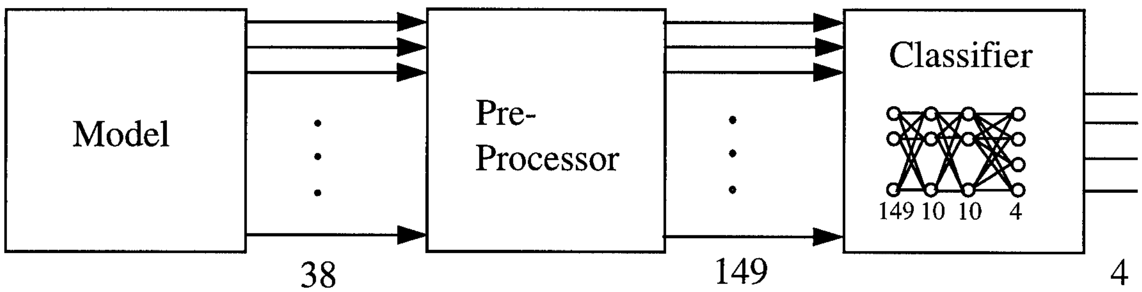


Figure 5 -1: Problem Structure

Network Configuration:

I first attempted a network structure consisting of 149 inputs, 38 hidden nodes, and four output nodes. The network converged with an accuracy of 88%, but it incorrectly classified each C1 unit. Of these errors, all but two contained outputs of 0000, thus indicating no preference. The network classified two C1 units as C2 for the remaining errors.

Increasing the number of hidden nodes to 76 compounded the problem. The network achieved no better than a 53% success rate and incorrectly classified all the C1 and C2 patterns. A hidden layer of fifty nodes returned similar results.

Reducing the number of hidden nodes from 38 reduced the training time but did not improve the accuracy of the network. Performance declined with less than 20 hidden nodes.

Next, I experimented with a second hidden layer. Using twenty nodes as a base from previous experiments, I achieved optimal performance with a network consisting of two hidden layers with ten nodes in each hidden layer. The network achieved 98% accuracy on the training data with 4 bit errors. The network incorrectly classified two patterns by one category (C1 to C2, and C3 to C2). Upon observation of the erroneous patterns, both errors were justifiable. Table 5-1 summarizes some of the results of the network configuration phase.

Layout	ETA	alpha	Bit Errors	% Correct
149-38- 4	0.9	0.2	19	88%
149-76- 4	0.9	0.2	66	53%
149-50- 4	0.9	0.2	66	53%
149-20- 4	0.9	0.2	17	86%
149-15- 4	0.7	0.1	18	87%
149-10-4	0.7	0.1	24	83%
149-12-8-4	0.7	0.1	10	94%
149-14-6-4	0.7	0.1	5	97%
149-10-10-4	0.7	0.1	4	98%

Table 5-1 - Results of Various Network Configurations

Generalization:

After configuring a network that could obtain acceptable results on training data, I performed a six-fold cross validation to test generalization. The 120 data sets were divided into six groups of twenty. The experiment was repeated six times with each of the six subsets used once as the test set, with theother five subsets used as the training set. The results are listed in Figure 5-2.

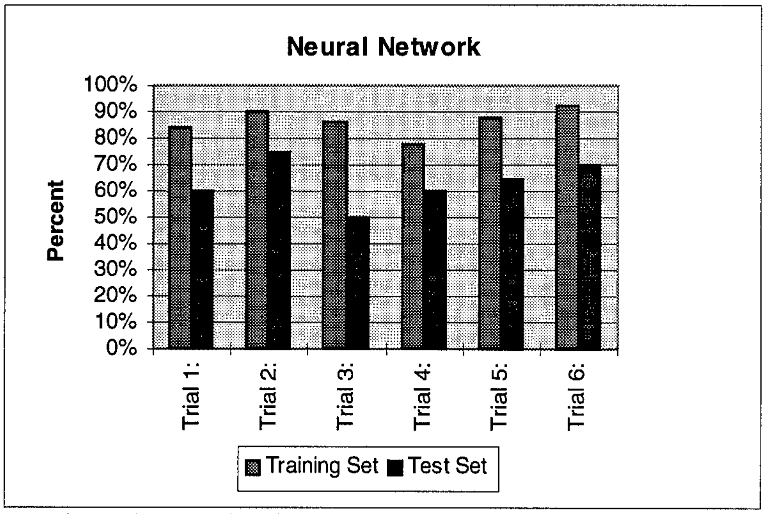


Figure 5-2: Results of the Back Propagation Classifier

Overall, 86% of the training patterns were correct with a 5% standard deviation, and 63% of the test patterns were correct with a 9% standard deviation over the six subsets. Table 5-2 shows the confusion matrix which is simply a matrix comparing the actual class to the neural networks classification.

NN→ Actual↓	C1	C2	C3	C4	Percent Correct
C1	2	11	2	0	13%
C2	4	46	2	0	88%
C3	3	12	4	2	18%
C4	2	2	3	25	78%

Table 5-2: Confusion Matrix for the Neural Network Classifier

It is apparent that the classifier was unable to identify C1 and C3 units. The reason is most likey the limited number of C1 and C3 training patterns available. As noted, it also requires more information to make a C1 determination, and there were limited examples from which to obtain this information. The fact that the matrix is not sparse indicates that there were insufficient examples. For the categories with the highest and second highest number of examples, the generalization results were 88% and 78%, respectively.

Weight Space Analysis

Battalion Commanders and Army Planners desire to know not only the capability of a combat unit, but they must know the reasoning behind the classification. For the 149-10-10-4 network, there were a total of 1,654 free parameters. A complete Hinton or Bond [Haykin, 1994] diagram does not adequately relate the network's reasoning process. While they do tend to show behavior by describing the weight space, they fail to capture the key features that produced the classification. Ideally, we would like to know these key features.

It is important to note that in the following analysis, I only considered the reasoning for an output node's excitation. The network is a *one-hot* system, meaning that only one output node is active per input pattern. One could also attempt to analyze why a unit was not assigned a classification based on the strength of the inhibitions; however, I examined the reasoning for excitation rather than inhibition.

With this goal, I attempted to examine the weight space in combination with the network state for a given input. First, I selected a pattern that both the expert and the network classified as C4. Examination of the pattern revealed three variables (category) that strongly indicated a C4 unit, eight variables indicating a C3 unit, twelve indicating C2, and the remainder indicating C1.

Figure 5-3 is a weight-space graph of the network's activation for this pattern. The Graphs indicate the percentage activation for each binary input. I included only those inputs (of the 149) that significantly contributed to the. Zero and negative valued inputs tended to prevent activation of a neuron while positive inputs facilitated the activation. Figure 5-3 shows that Neurons (1,2) and (1,3) contributed 28% and

72%, respectively, to the activation of the output neuron, (2,4) for the C4 example. The (1,2) notation refers to the 2^{nd} neuron in Layer 1. Layers are number from 0 to 2 with 0 being the input layer. Neuron (0,9) contributed 90% to the activation of (1,3), and Neuron (0,8) contributed 10% to the activation of Neuron (1,3).

From the figure it is possible to determine the primary bits responsible for the activation of the output neuron. Neuron (0,9) was responsible for 65% of the activation of the output neuron. Of this, Bit 110 was responsible for 18% or 12% of the output activation. Bit 59 was responsible for 12% of Neuron (0,9)'s activation or 8% of the output activation. Likewise, Bit 71 was responsible for 6.5% of the output activation. Bit 110 was a ground serviceability rating that is considered C2. Bit 59 is for a Training item in the C3 range. Bit 71 is for a Training item in the C3 range. Of the three items considered C4, two, Bits 28 and 64, contributed to the activation of Neuron (0,9) (total of 9% of output activation). Of the ten inputs contributing to the activation of node (0,9), five were C3 items. Since this node was primarily responsible for a C4 classification, I can only conclude that this combination of C2, C3, and C4 items suggested a C4 classification, and not a set of C4 specific items.

Second, I selected a pattern that the network correctly classified as C3. Figure 5-4 shows that it required a more complicated analysis to produce the C3 classification. This is justifiable because a higher classification requires that all the inputs be in a certain range. For example, if a unit had a high personnel and logistical classification but a poor training classification, it would receive a poor overall classification. In this sense, the classifier needs to verify that the unit meets all the

requirements before it assigns a high classification. On the other hand, only a few indications of poor performance is enough to justify a lower classification.

Bit 110 produced the strongest contribution to the output neuron contributing 11% to the output activation. This item would individually indicate a C2 unit, however. In fact, of the seven bits primarily responsible for the activation of Neuron (0,9), only one is tuned to a C3 indicator. This implies that a combination of features is required for this classification.

Also of note is Neuron (0,5). It contributed substantially to each of the neurons except Neuron (1,4). The features detected by this neuron were considered by three neurons in the next layer where Neuron (1,4) considered Neuron (0,9) almost exclusively.

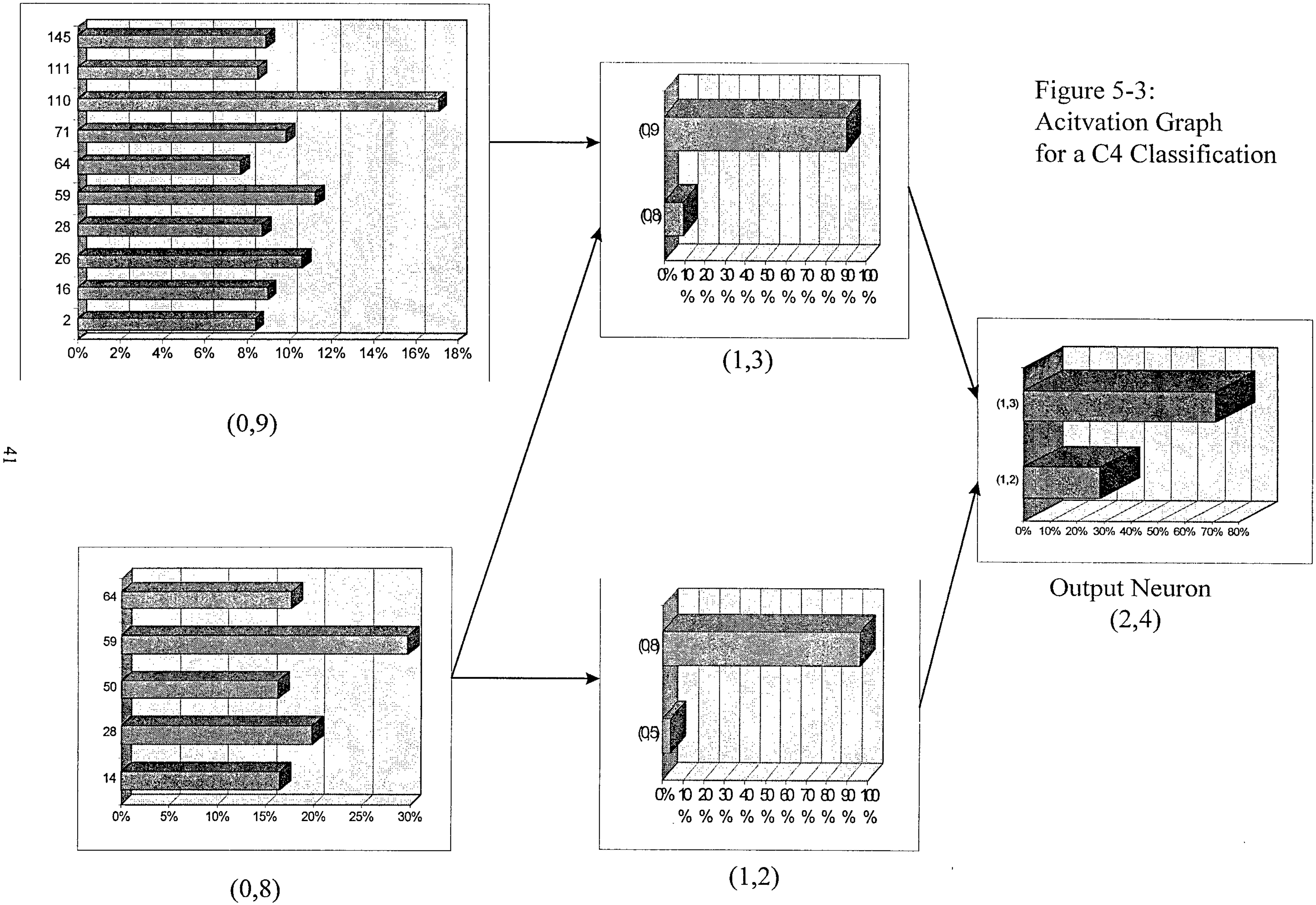
Conclusion

The neural network achieved an 86% performance rate on the training data with a 5% standard deviation and a 63% performance rate on the test data with a 9% standard deviation. The majority of the generalization errors occurred with C1 and C3 units for which there were limited training examples. For C2 and C4 units, the generalization percentages were 88% and 78%, respectively.

A neural network could be constructed with the combined expertise of various commanders and staffs and used to provide a consistent classification for Army units. However, neural networks have a major limitation. Because neural networks use a highly parallel approach to selecting combinations of features within an input set, it is difficult to determine the exact reasoning for a classification. A simple example

revealed the inherent difficulty in identifying key features. Additionally, it was shown that the network required a more complicated analysis of the data to assign units higher classifications.

Figure 5-3:
Acitvation Graph
for a C4 Classification



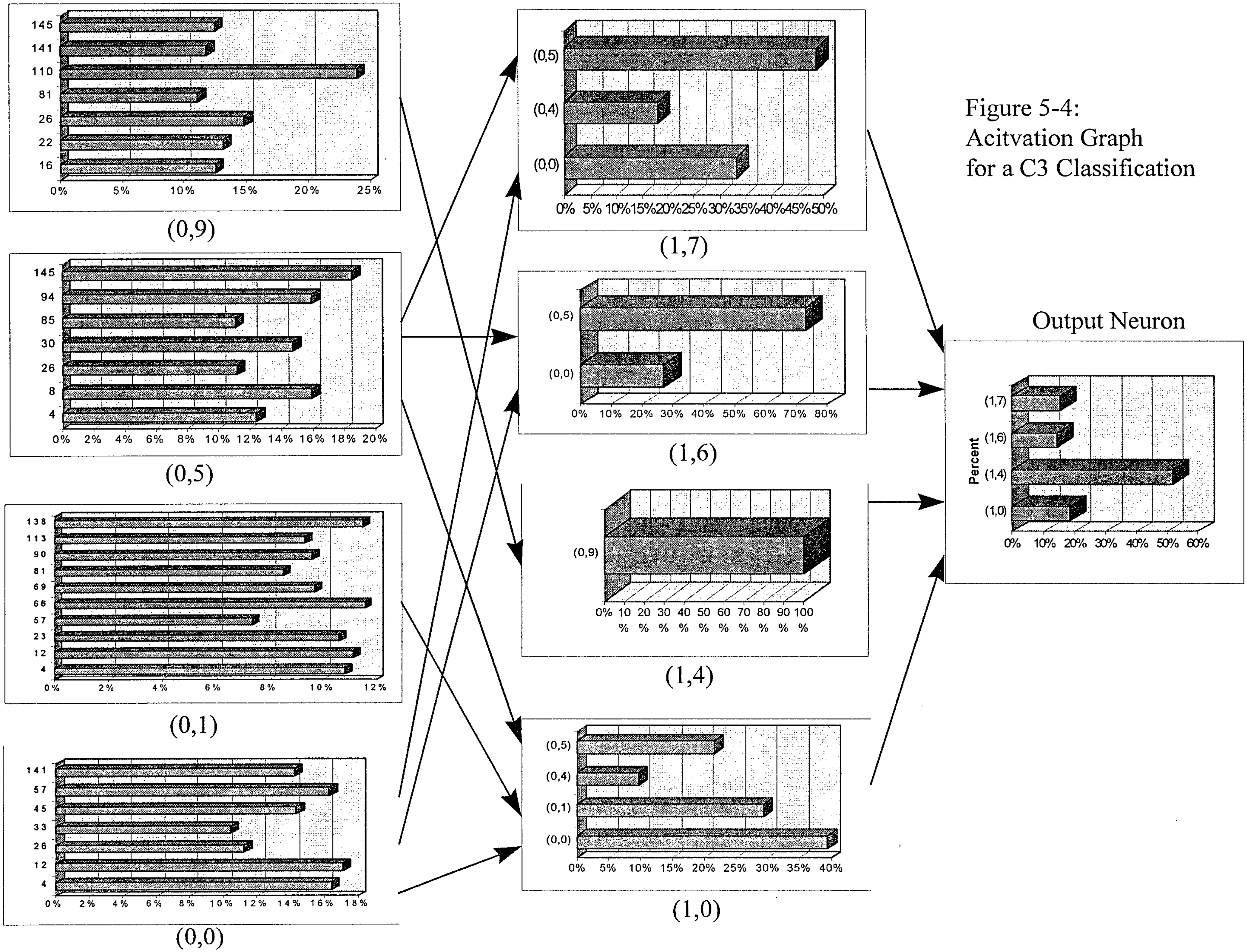


Figure 5-4:
Acitvation Graph
for a C3 Classification

Chapter VI

Decision Tree Induction

Introduction

Decision Tree Induction is quite simple, yet has been very successful for many applications [Michie, 1986], [Sammut, 1992]. The language of decision trees is propositional which means that rules comprised of conjunctions and disjunction's determine classifications. Figure 6-1 is an example of a decision tree.

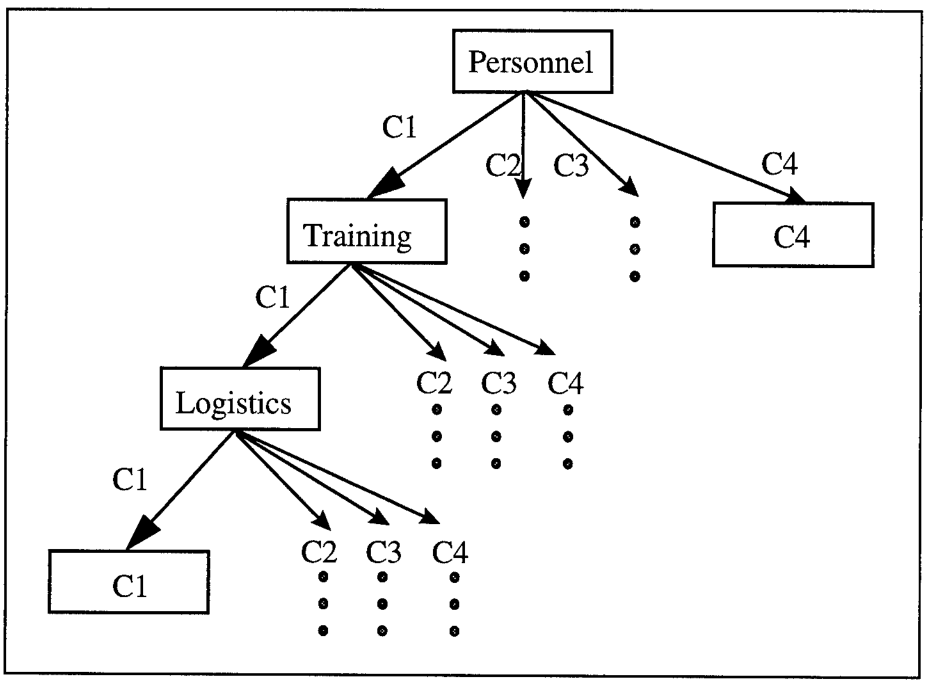


Figure 6-1: Sample Decision Tree

This tree has two visible leaves. One path begins with Personnel and follows a C1 arc to Training, then a C1 arc to Logistics, and finally a C1 arc to the leaf, C1. C1 is thus the classification for this data set. The equivalent propositional rule is

(Personnel Status = C1) ***and*** *(Training Status = C1)* ***and*** *(Logistics Status = C1) => C1.*

This rule has three conjuncts, and for each variable of each conjunct, there are four possible classifications (i.e., C1, C2, C3, or C4). This implies sixty-four rules for this simple, discrete example. The following rule might indicate C4:

(Personnel = C4) ***and*** *(Training = C1* ***or*** *C2* ***or*** *C3* ***or*** *C4)* ***and***

(Logistics = C1 ***or*** *C2* ***or*** *C3* ***or*** *C4) => C4.*

The latter sentence has three conjuncts and eight disjuncts and is equivalent to

(Personnel = C4) => C4.

The illustration is simple because the decisions at each node are discrete, and there are a limited number of them. As the tree grows and continuously valued variables are added, it becomes necessary to prune the tree and capture only the essentials of the decision making process. This eliminates any decisions that do not influence the final outcome. Induction learning looks at the examples and constructs a decision tree that can be used to make classifications. This tree can also be expressed as a set of rules.

C4.5

C4.5 is a machine learning algorithm that uses decision tree induction for classification. The code and a description of the algorithm is provided in [Quinlan, 92]. He states that a "decision tree can be used to classify a case by starting at the root of the tree and moving through it until a leaf is encountered...the class of the case is predicted to be that recorded at the leaf." C4.5 generates output in both "decision tree" and "rules" formats with accuracy data and data on the utility of the individual rules.

Data Representation

C4.5 was able to process the data in its original form. Discrete variables were inputted as alphanumeric quantities, and continuous variables were inputted as floating point numbers.

Results

The results of the six-fold cross validation are given in Figure 6-2. The mean for the six trials was 86% accuracy for the training sets and 67% for the test sets with a standard deviation of 5% and 9%, respectively. C4.5 generated only seven rules, and the most complex of these consisted of only five conjuncts. Clearly, the poor performance on the test sets is attributable to the limited number and complexity of the rules.

Not only were the rules overly simplistic, but they were not an accurate model. One rule indicated that no **more** than 82% of the unit could fire expert if the unit were to retain its C1 evaluation. Certain ranges for values were inconsistent within the rules, but there were several indications of success. It is possible that with many additional training patterns, the algorithm might produce rules that truly depict the nature of the decision rather than coincidental information.

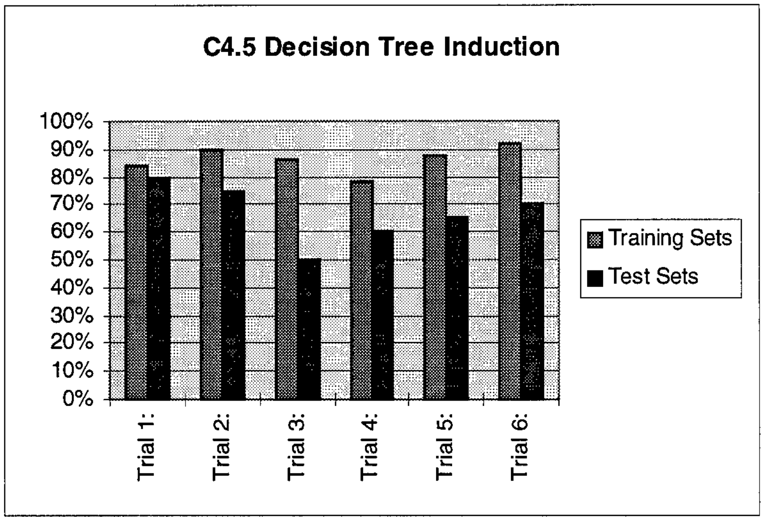


Figure 6-2: Results of Decision Tree Induction

It was suggested in Chapter 5 that it is easier to classify a unit with major deficiencies (C3 or C4) than one that has few or no deficiencies (C1 or C2). This is even more significant with decision trees. If an aviation unit, for example, only has 50% of its aircraft, then the unit is easily classified as C4. It is unnecessary to search further in the decision tree. However, if the unit has only a few deficiencies, then a complete search is necessary to ensure a C1 or C2 classification.

With 120 examples, C4.5 was unable to create rules that were sufficiently complex to classify C1 units. Only 66.7% of C1 training examples were correct, yet 90.6% of the C4 classifications were correct. This disparity is easily explained by the lack of depth in the decision tree.

Table 6-1 is a Confusion Matrix for the test cases using Decision Tree Induction. It is clear from the dispersion that the rules were not sufficiently complex to classify the C1 and C3 patterns. The classifier identified less than half correctly and inconsistent distributed the errors.

Actual→ Classifier↓	C1	C2	C3	C4	Percentage Correct
C1	7	6	1	1	47%
C2	5	41	4	1	80%
C3	2	6	9	5	41%
C4	0	3	11	18	56%

Table 6-1: Confusion Matrix for Decision Tree Induction

Decision Tree Algorithms apply Ockham's Razor [Russell, 95] which mandates the use of the simplest rule that correctly classifies the majority of the examples. However, this can cause poor generalization if the rules are overly simplistic. For the discrete Unit Classification Problem, there are over 7×10^{22} possible combinations. Surely, it is impossible to train on each of the possibilities, and ideally we would like to increase the number of partitions of each attribute which would increase the number of possible patterns.

Although there have been estimates on the required size of training patterns [Vapnik, 71], these estimates tend to be large, upper bounds. The best solution for minimizing the number of required training patterns is to incorporate known standards into the rule set and train on patterns that differ from the known standards. This approach could combine the benefits of expert systems with machine learning technology to assist unit classification.

Chapter VII

Simple Bayes's Classifier

Introduction

Another often used method is Bayes's Rule which is based in probability and statistics. Bayes's Rule states that

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)}$$

P(A I B) is the probability of an event A given the evidence B. P(B I A) is the probability of B given A. P(B) is the probability of B, and P(A) is the probability of A. For the Unit Classification Problem, we want to know the probability that a unit is C1 given its data, i.e.,,

$$P(Class \mid Data) = \frac{P(Data \mid Class)P(Class)}{P(Data)}$$

This formula will allow the computation of the probabilities of each class. The algorithm then assigns the pattern the classification having the highest relative probability. Since the probabilities are relative, the P(Data) is the same for each term and can be eliminated. The equation thus reduces to

$$P(Class \mid Data) \propto P(Data \mid Class)\ P(Class)$$

Assuming data independence, the P(Data I Class) can be expressed as the product of the individual probabilities, i.e.,,

$$P(Class \mid Data) \propto P(Data_1 \mid Class) * P(Data_2 \mid Class) * \cdots * \\ P(Data_n \mid Class)$$

for n data items. The algorithm then assigns the classification according to the highest relative likelihood.

The underlying assumption is data independence. Independence implies that the probability, $P(Data_i \mid C1) = P(Data_i)$, or equivalently $P(C1 \mid Data_i) = P(C1)$. This assumption might not be valid. For example, assume that a unit has an assigned personnel rating of 89%. This would suggest a C2 classification with a high probability. However, if the personnel available percentage were 100%, then the assigned shortage would not be as significant.

Algorithm

First, the discretized input space was compared to the expert's classification for each input. For each attribute, there were four ranges and four possible classifications. Each attribute, therefore, had 16 associated probabilities. Table 7-1 is a sample for the Assigned Personnel Attribute.

$Range_i$	$P(Range_i \mid C1)$	$P(Range_i \mid C2)$	$P(Range_i \mid C3)$	$P(Range_i \mid C4)$
1	1.00	.67	.45	.37
2	0.00	.33	.45	.22
3	0.00	0.00	.10	.25
4	0.00	0.00	.00	.16

Table 7-1: Probability Table for the Assigned Personnel Table

In Table 7-1, each column sums to 1.00. For each example that the expert classified C1, the Assigned Personnel Percentage was in Range 1. This indicates that the algorithm could not assign a classification of C1 unless the Assigned Personnel Percentage was in Range 1. For the examples that the expert classified C2, 67% were in Range 1 and 33% were in Range 2. For the C4 examples, the spread was evenly

distributed over the entire set of ranges. The entire probability table consisted of 593 probabilities.

For each pattern, the algorithm computed the probability of each class (1-4) using the probability table. There is a probability associated with each bit in the input pattern. For each bit, the corresponding probabilities were multiplied to form an aggregate probability. This product was then multiplied by the probability of the class to form the probability that the pattern belonged to the target class. The algorithm assigned the pattern to the classification having the highest probability.

Data Representation

The Bayes's Classifier used the discretized data described in Chapter 5. Each data set is represented by 149 bits corresponding to the 38 indicators. Each indicator has three or four bits, and the bit indicates if the indicator is in that range. For example, the raw data for a METL Task can either be T, P, or U. T would be represented as 1-0-0. P is 0-1-0, and U is 0-0-1 (see page 24).

Results

Figure 7-1 contains the results of the Simple Bayes's Classifier.

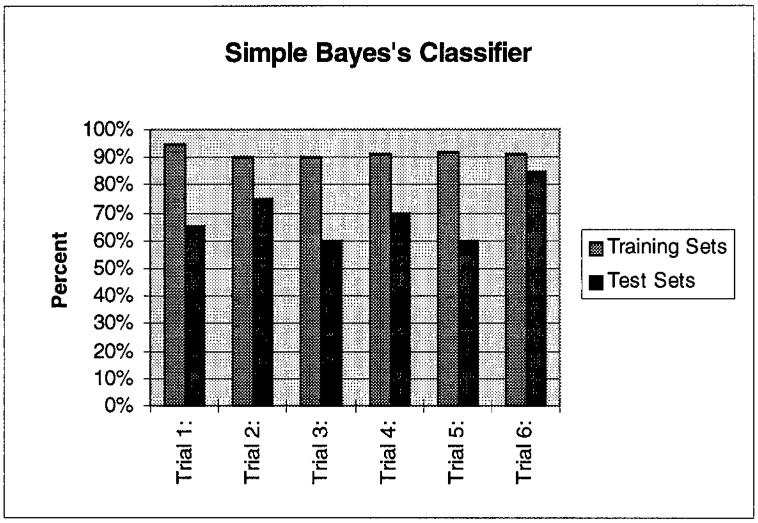


Figure 7-1: Simple Bayes's Classifier Results

For the six-fold cross-validation, the average percentage correct was 92% for the training data with a standard deviation of 2 %. The test set data was 69% correct with a 10% standard deviation.

Table 7-2 is a Confusion Matrix for the Simple Bayes's Classifier.

Actual→ Classifier↓	C1	C2	C3	C4	Percentage Correct
C1	4	10	1	0	27%
C2	2	44	4	1	86%
C3	0	5	9	8	41%
C4	5	1	0	26	82%

Table 7-2: Confusion Matrix for the Simple Bayes's Classifier

Table 7-2 shows that the C2 and the C4 classifications were most accurate. 86% of C2 items were classified correctly, and 81% of C4 items were accurate. However, the C1 and C3 items did not perform as well. Only 27% of C1 items were identified correctly along with 41% of the C3 items.

As discussed, classification of C1 items tends to be less accurate than the remaining classes as a result of the analysis. Additionally, the number of C2 (51) and C4 (32) patterns tend to improve their accuracy. The classifier identified 60 items as

C2. Interestingly, five C4 patterns (15%) were identified as C1. This is the most severe mistake that can be made, classifying units that are not ready for combat as having no deficiencies.

Conclusions

The Simple Bayes's Classifier classified 69% of the test cases correctly. The majority of errors were from the C1 and C3 patterns. As previously noted, C1 classifications tend to be more difficult because of the exhaustive evaluation of data. However, the limited number of C3 and C1 training patterns reduced generalization.

Chapter VIII

Nearest Neighbor Classifier

Introduction

Another simple method is classification based on the location of the input vector relative to an ideal class vector. The formula

$$distance = \sqrt{(x_1 - \overline{x_1})^2 + (x_2 - \overline{x_2})^2 \cdots (x_n - \overline{x_n})^2}$$

can be used to obtain the distance from the input vector to each of the average class vectors. x_n is the nth element of the target vector. Each element corresponds to a particular category of the unit, for example, a training, logistics, or personnel category. x_n bar is the nth element of the classification vector. There is a classification vector for each of the four classes. The difference between the input category and each classification vector is squared. The distance to the classification vector is the square root of the sum of the squared differences. The classifier assigns the input vector to the classification having the smallest difference, i.e., the nearest neighbor among the ideal class vectors.

Data Representation

The classification vectors are derived from scaled training data. The raw data was scaled and modified to fit a normal percentage (0 to 1.0). Each data set consisted of 38 scaled, continuous indicators. The C1 range was 0.9 to 1.0. The C2 range was 0.8 to 0.9. The C3 range was 0.7 to 0.8, and the C4 range was less than 0.7. Once again, I used a six-fold cross validation with 20 patterns in each test set. The

algorithm used the remaining 100 patterns to determine the four classification vectors by simply averaging each element over the 100 patterns based on its class.

Results

Figure 8-1 shows the results of the six trials. The classifier correctly identified 73% of the test patterns with a standard deviation of 11 percent.

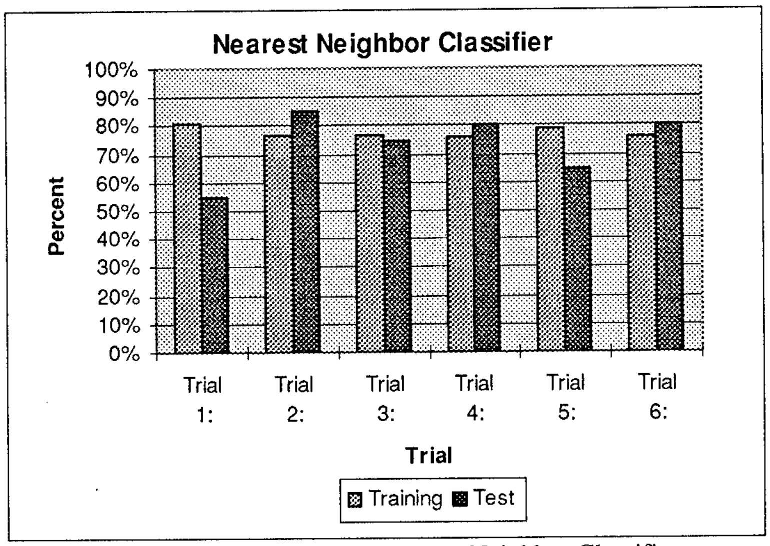


Figure 8-1: Results of the Nearest Neighbor Classifier

Figure 8-2 shows the list plot for the average classification vectors. As expected, C1 units tend to have higher values than the other units. However, there is a substantial amount of overlap among the C1, C2, and C3 units. In fact, the C1 and C2 curves are highly similar over the 38 categories. The C4 plot tended to be well below the other three, and this resulted in an 81% accuracy rate for C4 test patterns. Note that the C1, C2, and C3 patterns are centered at or above the 85% line. The data was scaled so that 80% to 90% would indicate a C2 unit, yet the C3 plot is centered at 85%.

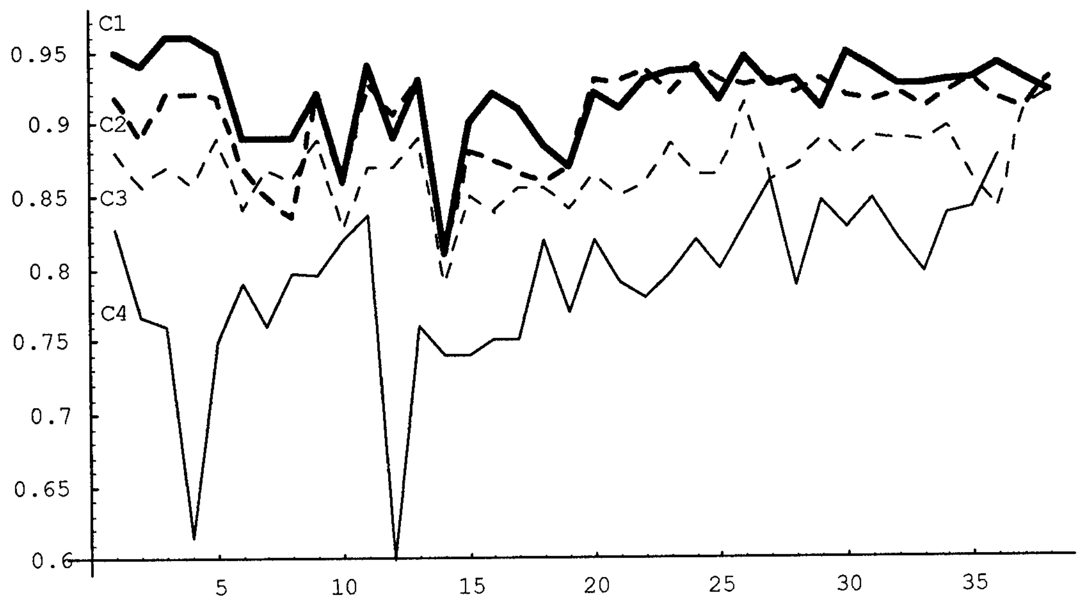


Figure 8-2: List Plot of the Average Classification Vectors

As expected, this approach resulted in a sparse confusion matrix (Table 8-1). The classifier incorrectly classified a C4 unit as C2, but this was the only test pattern to err by more than one category. This classifier correctly classified 80% of the C1 units and 68% of the C3 units. These figures are higher than previous results. The limited number of training samples was not as severe an obstacle to this algorithm because the samples provided a reasonable estimate to the mean; however, the classifier was only correct on 71% of C2 units.

NNC→ Actual↓	C1	C2	C3	C4	Percent Correct
C1	12	3	0	0	80%
C2	12	36	3	0	71%
C3	0	5	15	2	68%
C4	0	1	5	26	81%

Table 8-1: Confusion Matrix for the Nearest Neighbor Classifier

The majority of the C2 misclassifications were to the C1 category. This highlights a bias with which the nearest neighbor classifier has trouble. Overall trends

tend to support a particular classification, but individual categories are the basis for the final decision. The nearest neighbor classifier does not weight one category over another and does not attempt to distinguish between the categories. [Aha, 1991] states that nearest neighbor classifiers are (1) intolerant of irrelevant attributes, (2) intolerant of noise, (3) have trouble with nominal valued data, and (4) do not provide much knowledge of the structure of the data.

By normalizing the data, we can minimize problem (3) and Figure 8-2 does indicate the structure of the data. As discussed, however, irrelevant and relevant attributes are not distinguished. The low performance on the training set (78% correct) justifies this claim.

Conclusion

The nearest neighbor classifier identified 73% of the test patterns correctly, but only identified 78% of the training patterns. This approach is successful at establishing the trend of a unit, but as [Aha, 1991] argues, it can not separate the relevant from the irrelevant attributes. All categories possess the same level of importance in this classification algorithm, even though the expert weighted certain categories higher than others. Additionally, it does not possess the capability to filter noisy data.

Chapter IX

Trees

Introduction

Having examined the classification of 120 units using four types of classifiers, a natural question arises concerning the distinction of one particular class from another. The listplot of means in Figure 8-2 suggests that the class relationships are intricate.

One method of testing the similarity of classes is to use a tree classification scheme. Using the existing data, change the desired values to a one/zero format where one indicates that the current pattern is a member of the target class and zero indicates that it is not. Use the training set four times, once for each class. We can then construct a simple classifier that queries the target vector four times. For example, if the C1 query is positive then the algorithm returns C1. If the C1 query is negative then test for C2. The algorithm returns C2 on a positive match and checks for C3 on a negative match. If the C3 query is negative, then check for a C4 match. If the C4 result is negative, then the classification was inconclusive. Using the Bayes's classifier, decision tree induction algorithm, and Perceptrons, I constructed tree classifiers that followed this description.

Data Representation

The data representation varied according to the base classifier. The Perceptron Tree and the modified Bayes's Classifier used the binary representation described in Chapter 5, and the Modified Decision Tree used raw data.

Results

The performance of the tree classifiers closely imitated the performance described in earlier chapters. The Bayes's classifier had the best test results because of a 98% success rate on C2 patterns and 81% rate for C4 patterns. C1 patterns were recognized correctly 40% of the time while the C3 success rate was only 32%. Overall, 74% of the test patterns were identified using a six-fold cross validation.

The Decision Tree results were highly similar. The algorithm achieved the greatest success for the C2 (76%) and C4 (75%) patterns, with poor performance for the C3 (41%) and C1 (33%) patterns. There is evidence that the lack of C1 and C3 patterns was responsible for the low success rates. In fact, the trained C1 classifier was 88% correct when given a test case. It tended to return negative for every pattern. While this method achieved a high success rate individually, it was highly ineffective in the tree context. The 105 negative/ 15 positive balance simply was not sufficient to produce adequate rules.

Finally, I constructed a Perceptron tree algorithm taking advantage of the preprocessed data described in Chapter 5. Perceptrons are valuable only for problems

that are linearly separable, so we would not expect the results to be very good. In fact, they were only slightly lower than previous results. By taking advantage of domain knowledge, certain weights can be fixed to look for certain attributes. When a key attribute is in a given range, then a fixed weight can highly influence the classification. Using fixed weights, the Perceptron tree algorithm identified 81% of C4 patterns and 76% of C2 patterns. Once again, C1 and C3 patterns were lower at 40% and 27%, respectively. Figure 9-1 summarizes the tree classification results.

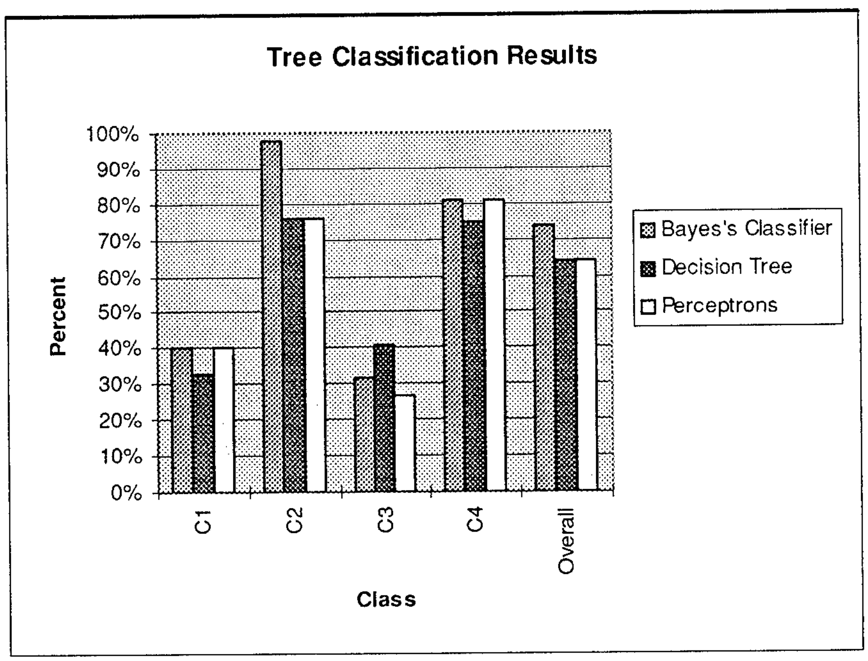


Figure 9-1: Tree Classification Results

Conclusions

A binary tree classification approach did not improve the performance characteristics significantly. Although the individual classifiers were able to obtain high success rates, when combined in a tree classifier, the overall performance was

similar to the aggregate classifiers. Perceptron trees achieved modest results despite their linearity limitation. This limitation was overcome using preprocessed data and fixed weights. Using raw data and unconstrained weights, the Perceptrons were unable to converge. No classifier was overwhelmingly superior in overall performance, but the Bayes's Classifier did maintain a slight performance edge.

Chapter X

Comparison of Classifiers Using Selected Variables

Introduction

In previous experiments, the assumption has been that the data set was complete and noiseless. In reality, units will often have both incomplete and noisy data from which to make their assessments. Often the noise will be transparent because it will be hidden deep within a variable. For example, a clerk might enter an erroneous value for a single piece of equipment that alters the overall equipment serviceability. A company within a battalion might be on a training exercise during the report, and some of their data might not be available. In fact, one of the arguments against an automated evaluator is that the commander and staff "know" the unit and can compensate for data inaccuracies. This chapter explores the performance of the classifiers using incomplete data sets. Noise is not specifically addressed because it is an inherent part of the data that is impossible to distinguish. In fact, the statistical nature of the modeling algorithm mitigates problems associated with noise.

I chose three, six, and nine variables from a total of thirty-eight in the complete data set. For the set of three, there is one item from each of the three areas: personnel, training, and logistics. For the set of six, there are two from each area; and for the set of nine, there are 3 from each area. I chose the variables that demonstrated the best separation of the data given two main indicators. The first indicator was

correlation of classification and variable, i.e., how closely the unit classification varied with the value of the variable. The second indicator was the separation provided by the normalized means. Figure 8-2 shows a continuous plot of the normalized means. The selected variables provided the maximum separation between classes.

Results

Figure 10-1 shows the classification results when using three, six, and nine variables together with the total results. All the classifiers obtained their lowest success rates when using three variables. The Bayes's Classifier correctly classified 75% of the test patterns when using only six variables while the average for the other three was 63%. When using nine variables, the Neural Net was successful on 84% of the test patterns, exceeding the other three classifiers.

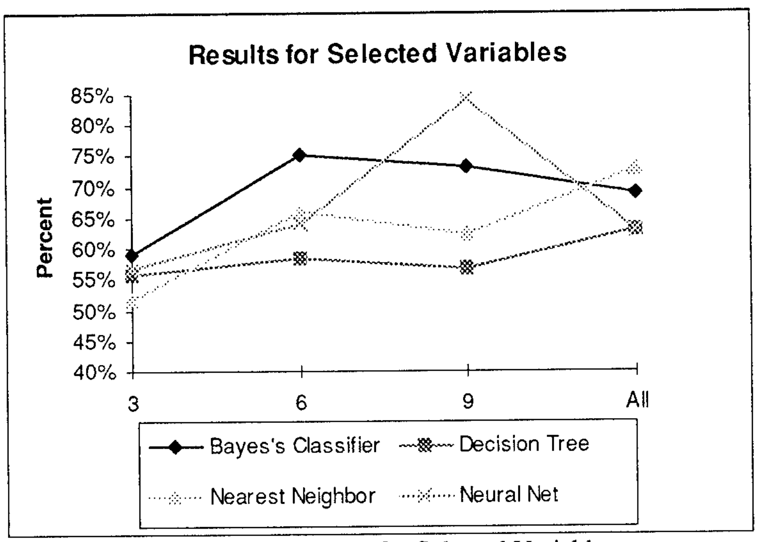


Figure 10-1: Results for Selected Variables

Perhaps more interesting is the inconsistencies relating to the order and deviation of the classifiers. It is apparent from Figure 10-1 and Figure 10-2 that no one classifier is inherently better suited to this task. Rather, they depend on the

information available. Success rates had a much higher deviation with six and nine variables compared to the extremes of three and thirty-eight.

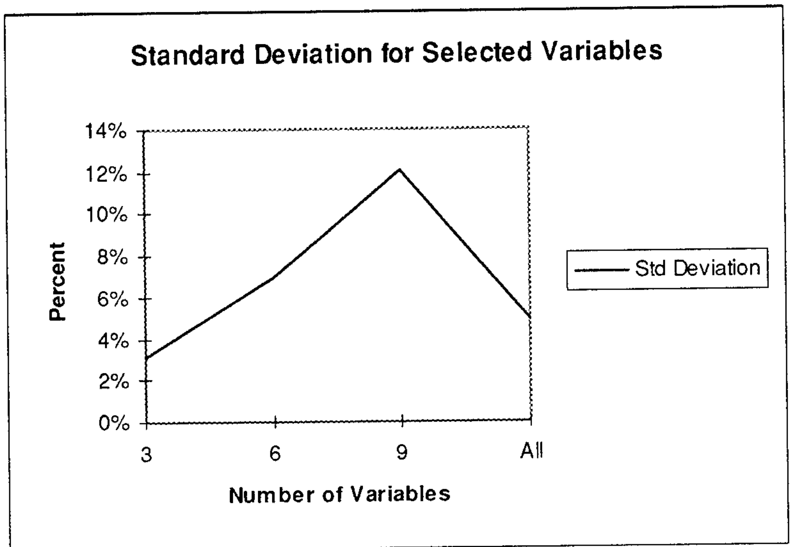


Figure 10-2: Standard Deviation for Selected Variables

Additionally, some classifiers performed better on particular classes. Figure 10-3 shows the combined results of the three, six, nine, and total experiments for each class. The Nearest Neighbor Algorithm far exceeded the norm for classifying C1

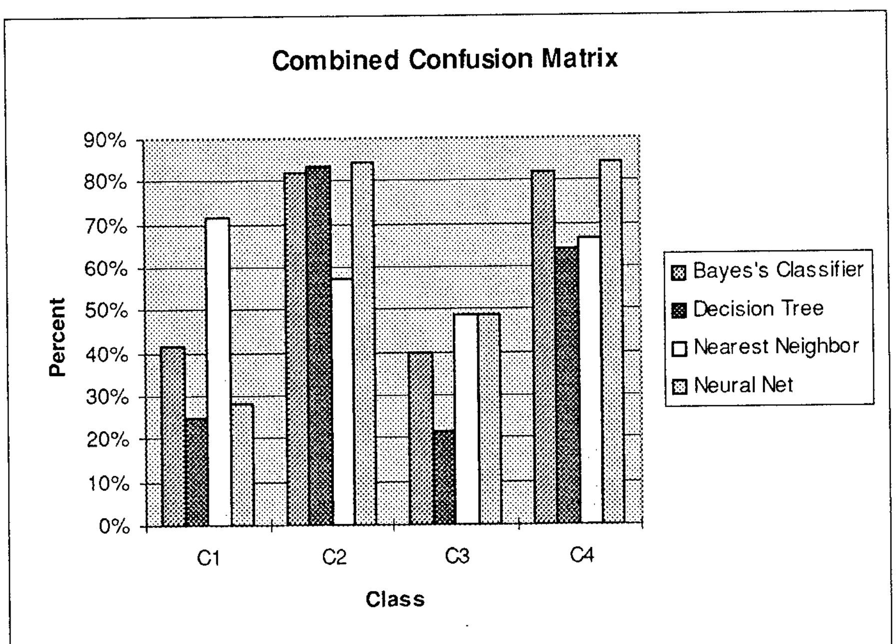


Figure 10-3: Comparison of Classifiers Combined Performance Per Class

units. Despite lower than average rates for most classes, the Decision Tree algorithm performed very well on C2 units. The success rate for C3 units was below 50% for each classifier, and the Neural Net achieved the highest rate for C4 units. Overall, the Bayes's Classifier had the highest average rate. The preceding demonstrates the inconsistencies of the algorithms.

Conclusions

In this chapter, I examined the performance of the four classification algorithms using reduced data sets. Using three variables and the complete set of 38, the standard deviation for the classifiers was much lower than for six and nine variables. The variation and the lack of consistent ordering indicated that no one classifier was best suited to this classification task. C2 and C4 units experienced much higher classification rates on average, but the Nearest Neighbor algorithm achieved the most consistent success rates.

Chapter XI

Conclusion

The original problem was to automate the analysis of the information in the battalion and provide an intelligent judgment as to the status of the individual areas and the overall readiness of the unit. It would be an easy problem if readiness were clearly defined. Unfortunately, evaluating readiness is a difficult problem that requires experience and judgment. The task is, therefore, to capture this experience and judgment. To simulate a unit, I created a model consisting of 38 indicators that represented the data in a battalion. An Army Lieutenant Colonel evaluated the combat potential of 120 of these *units* based on the data. He agreed with the *hard* rules listed in AR 220-1 for 74% of the *units*. For the others, he used his experience and judgment and ability to combine a combination of indicators to either upgrade or downgrade the readiness potential. Experiments were conducted using neural network, decision tree, Bayes's Classifier, and nearest neighbor algorithms. There were several trends among the algorithms:

(1) Performance varied from 63% to 84% on previously unseen data. No single classifier was significantly better than the others.

(2) Performance on units classified as C2 and C4 was vastly superior to units classified as C1 or C3. This corresponds to the number of training samples available.

The nearest neighbor algorithm achieved the best success rates for the C1 and C3 classes.

(3) Performance did not decline when the number of indicators was reduced significantly. Two of the classifiers showed increased performance with a reduction of indicators. The standard deviation in the performance was lowest when all the indicators were available.

(4) Two of the classifiers required preprocessed data. The preprocessing did not alter the original data, but it did incorporate preexisting knowledge of the domain.

(5) A requirement exists to justify the reasoning behind a classification. The reasoning used by neural networks is more difficult to discern than the other algorithms.

The technology exists to automate the classification of Army Battalions, and the equipment to automate the collection is in the final stages of fielding. However, there is a majority of individuals who would argue that we should not take the evaluation process away from the commander. He has the *feel* of the unit, knows the soldiers, and can see the equipment. Although most of the objectives of the USR might be met without a commander's insights, surely commitment to battle necessitates his judgment.

What the technology offers and what Force XXI forecasts is not replacing the commander's judgment but creating an environment that enhances his abilities in real time. Intelligent daemons can constantly retrieve and extract information from the personnel, training, and logistical databases, updating assessments as the data changes.

Knowledge Discovery algorithms (Appendix B) can combine the data to produce previously unknown conclusions. The commander can have a natural language interface for complicated queries such as maintenance prediction based on current operating conditions. This status can then be uploaded to higher echelons to update their databases for use by their intelligent algorithms. These daemons would operate in the background constantly updating and advising thereby reducing the requirement for excessive staff personnel and increasing the number of combat personnel. This information would be mobile, not subject to the upheaval caused by the movement of the operations center. This paper demonstrates the potential for intelligent algorithms using current machine learning technology in the domain of readiness classification.

Appendix A

Unit Modeling Algorithm

Purpose

I elected to generate data rather than use existing data for two reasons. First, existing data has a confidential classification. This is not an insurmountable problem, but confidential information does restrict dissemination and collaboration. Second, existing data would not properly cover the data spectrum. Most units would center around C2 with a few low end C1 and a few C3. Very few C4 data sets would be available. The problem is that if we do not expose the learning system to the entire range of potential inputs, it may classify a pattern obviously located on the extremes of the output set incorrectly.

But does generated data lose the knowledge contained in actual data? Certainly, there are input values that are strongly correlated. Examples include the senior grade available percentage from the personnel data set and the leadership training rating from the training data set. Experienced leaders tend to be more highly trained, and this correlation was included in the algorithm. Most likely, there were correlated categories that were not properly modeled. It is, in fact, unlikely that we could define all the relationships between data elements. This is an inherent complexity in the problem.

Is this a limitation? I do not believe so because the expert had complete visibility of the data, and was able to judge the perceived inconsistencies. In fact, he

could use these inconsistencies to improve or reduce his rating. This is what a field commander might do if he realizes a low score in Physical Training was due to excessive field time over the last six months and not the overall physical condition of his soldiers.

This raises the issue of whether all the relevant data is included. Certainly, all the information specified by AR 220-1 is included with the exception of funding issues, training facilities, etc. These areas address reasons for conditions rather than the condition of the unit, so they were intentionally omitted. The question is whether essential unit indicators were omitted, and it is a question that is difficult to answer. Many argue that cohesion, espirit, teamwork and other intangibles are essential to unit evaluation. I would only comment that, if valid indicators, these should translate into measurable areas such as physical training, marksmanship, and maintenance.

Algorithm

The unit modeling algorithm produced 120 units consisting of 38 individual data elements. The data elements consisted of five for personnel, eighteen for training, and fifteen for logistics.

The algorithm consists of four major components as listed in Table A-1. The algorithm first randomly biased the unit toward a particular rating. The objective was biases consisting of fifty percent C1, twenty percent C2, twenty percent C3, and ten percent C4. The bias established a range for the indicators, but within the range the indicator received a random value. This design is based on actual practice, and correlates to certain units receiving higher priority for resources than other units. For

example, certain units are designated to maintain a high readiness rating. These are rapid deployment units, and receive a high priority on their resource requests. A bias of C1 corresponds to this priority. Other units receive a lower priority for their request. These percentages evolved as I attempted to achieve a normal distribution.

Table A-1: Unit Modeling Algorithm

```
For i =1 to 120 do
  begin

     (1) Determine the bias for the unit.

     (2) Based on the bias, generate the personnel data and normalize it.

     (3) Based on the bias, generate the training data and normalize it.

     (4) Based on the bias, generate the logistics data and normalize it.

  end;
```

A normal percentage consisted of a value between 0 and 100. If a unit had a C1 bias, then the normal percentage was randomly distributed between 85 and 100. A C2 bias produced a normal percentage in the range of 75 to 100, and so forth for C3 and C4. Table A-2 displays the probability distribution of the normal percentage as a function of the bias. The first column shows the bias. The second column shows the range that an indicator measured in a normal percentage can be assigned. In other words, if a data set has a C1 bias, then an indicator measured in a normal percentage is randomly assigned a value in the range 85-100. The normal range for C1 is 90 - 100, so the probability that the value is C1 is 10/15 or 66%. The probability that the value is in the C2 range is 5/15 or 33%.

Table A-2: Bias Probability Distribution

Bias	Range	Probability C1	Probability C2	Probability C3	Probability C4
C1	85-100	66%	33%	0%	0%
C2	75-100	40%	40%	20%	0%
C3	65-100	29%	29%	29%	14%
C4	55-100	22%	22%	22%	33%

For the categories that were not evaluated with a normal percentage, similar strategies forced ratings towards this distribution. For the METL task example, the domain consisted of T for trained, P for partially trained, and U for untrained. The distributions were randomly assigned according to the strategy in Table A-3.

Table A-3: METL Task Distribution

Bias	Probability T	Probability P	Probability U
C1	70 %	20 %	10%
C2	65 %	20 %	15 %
C3	60 %	20 %	20 %
C4	20 %	40 %	40 %

The normalization process employed a simple scheme to enforce data dependencies. For example, the available strength percentage can never exceed the assigned strength percentage. The probability that a unit is trained in a METL task is correlated to the months since certain training events. The algorithm attempted to resolve these dependencies. However, it is dependent on the heuristic evaluation of the dependencies. I am confident that not all the dependencies are known.

As stated, the expert evaluated 120 units. The results of his classification are listed in Table A-4. His evaluations did not fit a normal distribution, because he

emphasized key features within the model. For example, he considered a U in the training data highly significant which tended to skew the distribution to the right.

Table A-4: Distribution of Unit Modeling Data

Rating	Number	Percentage
C1	15	13%
C2	51	42%
C3	22	18%
C4	32	27%
Total	120	100%

Appendix B

Description of Terms

Multi-Layer Perceptrons With Backpropagation

Biological Neurons form the basis for artificial neurons. A typical biological neuron is illustrated in Figure B-1. A comparison of Figure B-1 and Figure B-2 reveals the similarities between the two models. The artificial neuron has inputs, weights on the inputs, a summing component, a nonlinear mapping function, and an output. The biological neuron has similar components: dendrites (inputs), soma (summing component), axon hillock (nonlinear mapping function), synapses (connections or weights) and an output (axon).

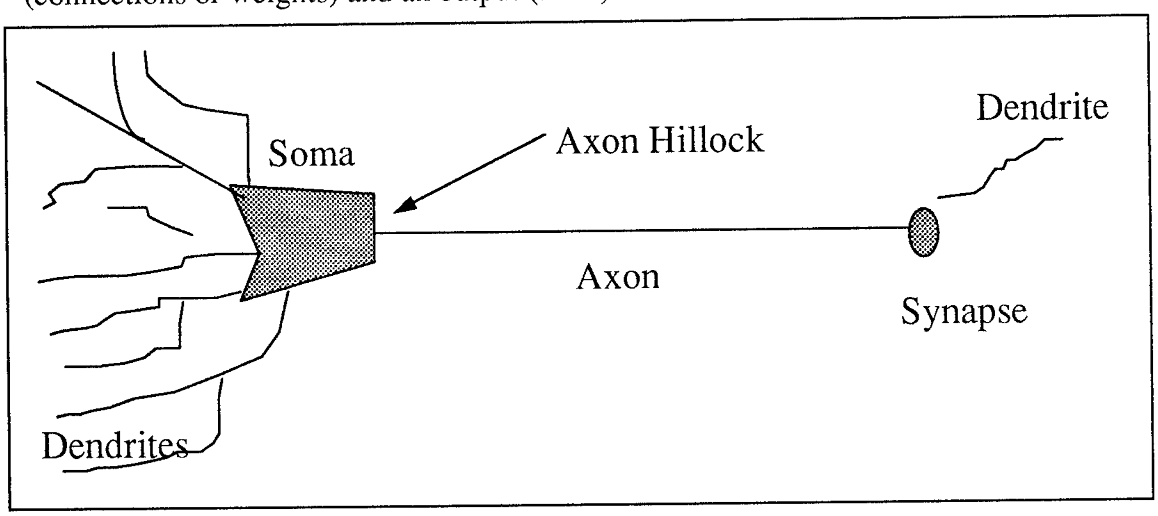


Figure B-1: Neuron

Neural Network research began in 1957 with the invention of the Perceptron by F. Rosenblatt. In its simplest form, the Perceptron is a linear filter that maps a

series of inputs to a single output. Figure B-2 is a graphical model of the single element Perceptron.

There is a problem with Perceptrons, however, that limits their usefulness. They are only useful for problems that are linearly separable. A class is linearly separable if all members of the set can be placed into one of two categories. Figure B-3 is an example of both a linearly separable problem and one that is not linearly separable. Most interesting problems tend to have more than one class. We typically want to classify things in more complex ways than just good/bad or yes/no, for example. In 1969, M. Minsky and S. Papert published a book that denounced Perceptrons because of this limitation. This effectively halted neural network research until the 1980's.

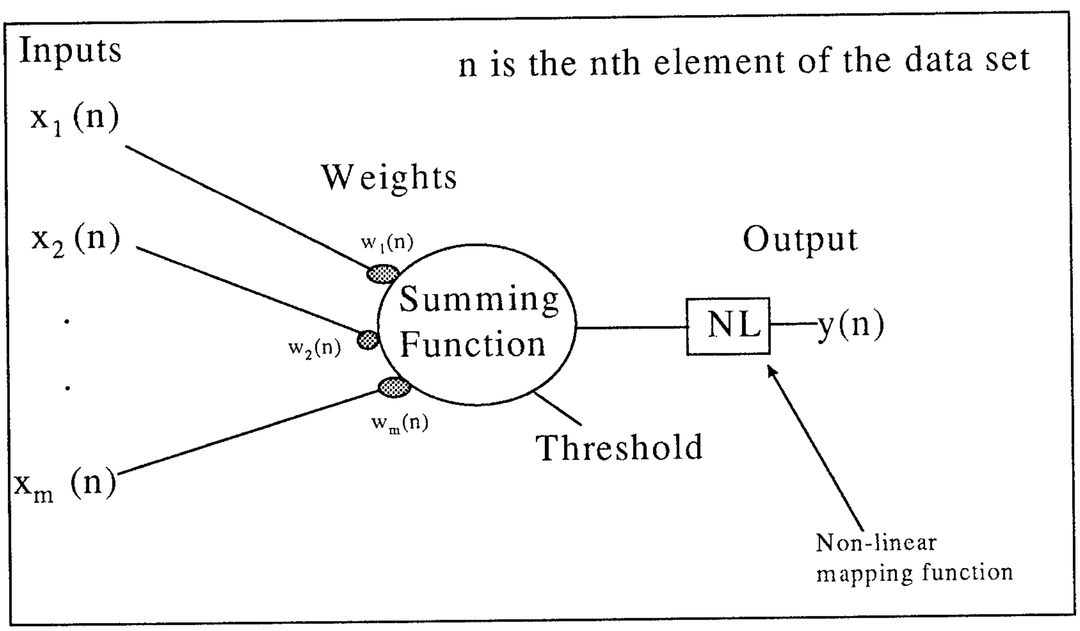


Figure B-2: Perceptron

In the mid 1980's, researchers led by D. Rumelhart and J. McLelland [Rumelhart, 86] standardized a method of training multi-layer Perceptrons. Figure B-

4 depicts a multi-layer Perceptron where each neuron is similar to the model in Figure B-2. These neurons are arranged in layers with an input layer, one or more hidden layers, and an output layer. Once again, there is a biological basis for this construction. Humans have neurons that receive sensory inputs, billions of interconnected neurons that process the inputs, and neurons that relay information to muscles. In fact, it is this massively parallel nature that provides the power of human intelligence.

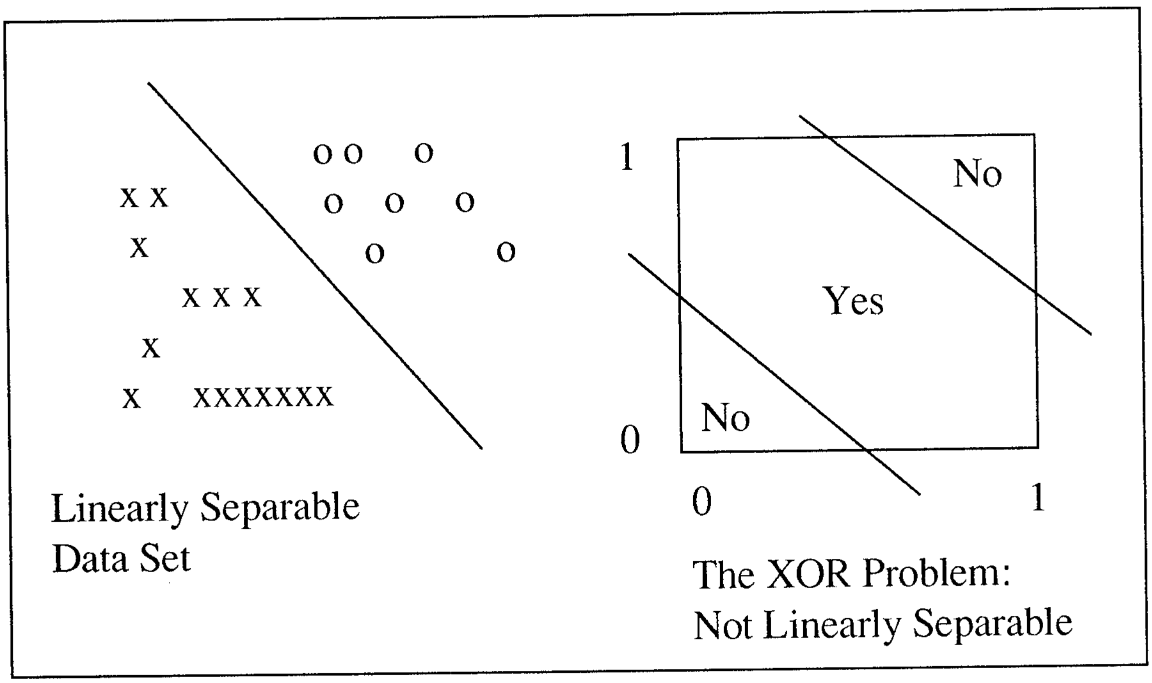


Figure B-3: Linear Separable Data Sets

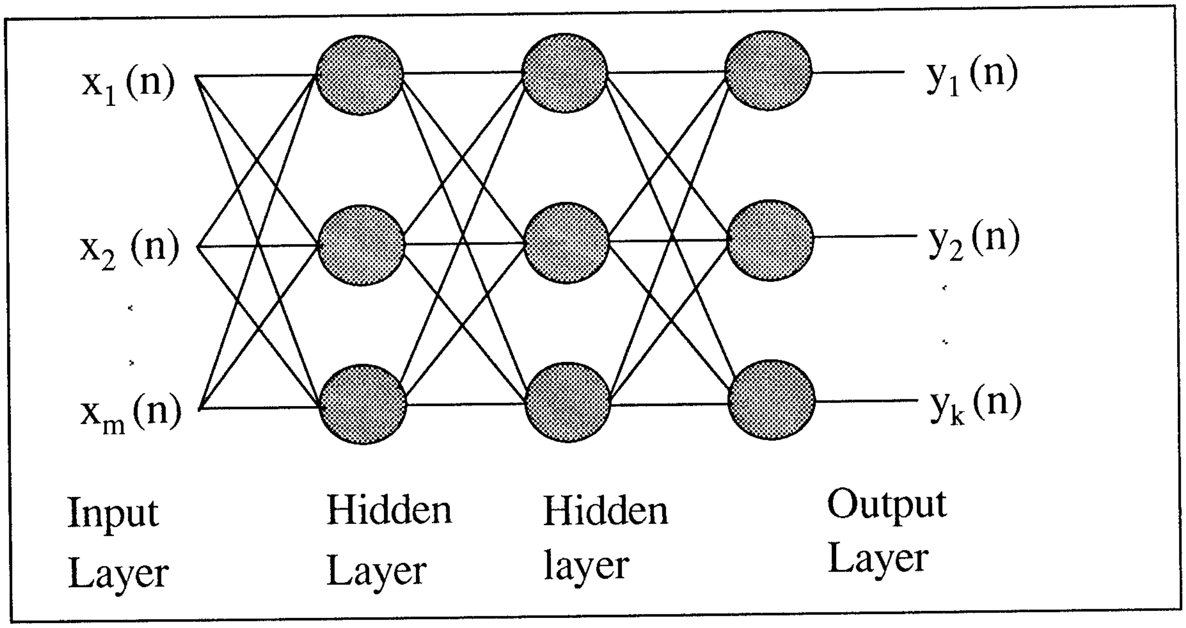


Figure B-4: Multi-Layer Perceptron

In its typical role, the multi-layer Perceptron is a vector mapper. It takes an m-dimensional input, processes it, and produces a k-dimensional output. Typically, k is much less than m, so that you are taking a complex set of inputs and returning a classification for those inputs. A typical problem is the XOR problem. The XOR has two input bits with four (2^2) total inputs and one output bit so two potential outputs. Table B-1 is an example of the XOR problem. The multi-layer Perceptron maps 00 and 11 to 0, and it maps 01 and 10 to 1. By increasing the size and complexity of the multi-layer Perceptron, we can increase the problem solving potential.

(n)	Exemplar	$x_1(n)$	$x_2(n)$	y(n)
1	00	0	0	-1
2	10	1	0	1
3	01	0	1	1
4	11	1	1	-1

Table B-1 - Representation of the XOR Problem

The Backpropagation algorithm as presented in [Haykin, 94] is given in Table B-1, and Table B-2 provides a summary of the relevant taxonomy.

1. Initialization: Determine the number of layers and neurons in each layer. This determines the number of weights. Set each weight to a small, random value.
2. Presentation of Training Examples: Present the network with each pattern in the set of training examples. For each pattern, perform steps 1 and 2.
3. Forward Computation: Each neuron in the input layer processes the input, and computes an output. The output of each neuron in the input layer is passed to each neuron in the first hidden layer. This continues to the output layer, and the output of the output layer is the output of the system.
4. Backward Computation: Compute the error of the output layer. Use this error as a guide for updating the neurons in the output layer. Then, propagate this error back through the network based on the contribution of the target neuron.
5. Iteration: Compute the sum of squared errors for each pattern in the input set. Repeat the above process until the sum of squared errors is sufficiently small.

Table B-2: Error Backpropagation Algorithm

Neural Network	Interconnected neurons arranged in layers trained to produce a desired output when presented with a specific input.
Neuron	Basic element of the neural network. Contains dendrites, soma, axon, and synapses.
Dendrites	Inputs to the neurons.
Soma	Computational compartment of the neuron. Multiplies the input with the associated weight, computes the sum of these products.
Axon	Output of the neuron
Synapse	The connection between neurons, a weight.
Exemplar	The set of input/output pairs. For each input, there is a corresponding output. The total set of pairs forms the exemplar.

Table B-3: Taxonomy for the Backpropagation Problem Domain

Database Systems

[Wiederhold, 83] defines a *database* as related data, the hardware that stores the data, and the software that manipulates it. [Date, 95] prefers "a systematic methodology for the standardization and integration of data resources at the organizational level," and [Frawley, 93] describes a database as "logically integrated collection of files." For the purposes of this paper, we will assume the latter, more limited definition. Specifically, database will refer to a relational database consisting of persistent data (extension) and an associated data dictionary (intension) that specifies the data types, field values, ranges, and related information.

A *distributed database* is simply a database where the data is stored on more than one node. A node can be a separate workstation with its own secondary storage, a processor with little secondary storage, or possibly a unit of secondary storage with only enough computational capacity to retrieve and store data. The nodes are connected via modem or a network. In contrast, a *centralized database* is a single database residing on the same node. Also, distributed database are homogeneous.

The term Heterogeneous Databases can have different meanings. Heterogeneity can refer to differences in database systems, operating systems, or the hardware it runs on [Sheth, 90]. For clarity, databases that operate under different operating systems, utilize different database management systems, have different query languages, or operate on different hardware platforms are considered heterogeneous.

A *deductive database* or *logic base* uses logic and logic based programming to extend the capabilities of the database [Sirounian, 95]. It consist of facts and rules. The facts can be related to the extension of a database, and the rules are similar to the intension.

Knowledge Based Systems

A knowledge base represents facts about the environment [Russell, 151]. It is similar to database systems in many ways. The knowledge base contains a series of representations or facts. The knowledge base administrator (automated or human) adds, updates, or removes facts as the environment changes, and users search the knowledge base for information about the environment. The information in a knowledge base differs somewhat from that in a typical database. The knowledge base contains facts about the domain, but the facts are expressed in a knowledge representation language.

A knowledge representation language should be unambiguous, clear, concise, and efficient. First order logic is a typical base for representation languages. It also has the advantages of being widely studied and well defined. The basic elements of first order logic are as follows:

Element of First Order Logic	**Description**
Connectives	And, or, implies, equals Example: (A and B) is true if both A and B are true
Quantifiers	For each, There exists Example: For each X there exist a Y such that if X is true then Y is true.
Constants	A nonchanging object in the world Example: HEMMT
Variables	One of a set of objects.

	Example: X
Predicates	A description of a variable. Represented as a tuple. Example: TRUCK(HEMMT, 5 TON, 2.5 TON)
Functions	Relates one variable to another Example: classification(HEMMT) = TRUCK
Sentences	Logic expression that represents a fact Example: (A and B or C)

Table B-4: First Order Logic

A sentence is the basic structure stored in the knowledge base. The fact that sentences are based in logic provides the power of the knowledge base and distinguishes knowledge bases from databases.

One of the major goals of the DARPA Knowledge Sharing Effort is to standardize the representation knowledge in knowledge based systems. [Patil, 95] notes that application specific representations are necessary, but describes a language that could be used as an interchange format. The Knowledge Interchange Format is an extended version of First Order Logic that is designed to be the basis for libraries providing reusable components. KIF is also intended to allow the interchange between application specific domains. The sending system would translate its specific representation into KIF, and the receiving system would then translate the KIF into its internal representation. An example KIF sentence is

```
(defrelation HEMMT (?x) :=
    (and (truck ?x) (ten-ton ?x)))
```

which would indicate that a HEMMT is a ten ton truck. As this example demonstrates, the sentences are contained in lists and have a linear, ASCII syntax.

The second component of a knowledge based system is the inference mechanism. This automated component is made up of rules that are used to control how the rules in the knowledge base are used or processed. It directs operations by deciding which sentences are applicable, how they should be applied, when enough sentences have been considered, and what possible solution is implied [McGraw, 4]. This inference mechanism is usually based in one of four language representation categories.

Information Extraction

Information Extraction (IE) Systems analyze text-based data and return relevant facts from the data. The systems do not attempt to understand all of the text, but rather attempt to determine the relevance of specific passages based on inherent knowledge of the query. The product of an IE is, ideally, a database of entries relevant to the problem. Designers pre-determine the database's column headings, and insert information descriptors obtained from the text search. Often, cells are best filled with strings from the source text.

IE systems must have the ability to perform limited natural language processing. They must be able to perform word recognition and sentence analysis, and be able to understand the subject of the overall document. Dictionaries are usually tailored to the problem domain to better support the abbreviations, technical terms, names, and jargon specific t the domain.

Many fields are currently developing IE systems. Some examples are Health Care, Intelligence Gathering, technical literature monitoring, and intelligence

gathering. Researchers are developing IE systems in Health Care that summarize medial patient records, assist with quality assurance studies, and support insurance processing. Many technical companies are interested in developing databases of current technology in order to stay ahead of competitors. IE systems automate this process by analyzing the relevant publications. A final example is government and business organizations that monitor newswire and on-line documents for intelligence gathering. Terrorism prevention and industrial competition are sample applications.

[Lehnert, 95] describes two metrics useful in assessing the performance of IE systems, recall and precision. Recall refers to "how much of the information that should have been extracted was correctly extracted," and precision is described as the "reliability of the information extracted." In studies conducted at the University of Massachusetts, humans exhibited 79% recall and 82% precision on information extraction tests. Automated systems achieved 53% recall and 57% precision indicating that IE systems are not currently as capable as their human counterparts. However, automated Systems have a much greater throughput.

Information Retrieval

In Information Retrieval (IR), the task is to choose from a set of documents the ones that are pertinent to a query. Early systems used Boolean connectives to search through keywords or abstracts to find a document matching the query. So much text is now available on-line that systems have become more sophisticated. It is now more common to search the entire text rather than abstracts, and vector-space models have replaced Boolean connectives [Russell, 95].

The vector-space model considers every list of words to be a vector in n-dimensional space. This includes both the query and the target text. It then compares the query vector against the target vector and reports the ones that are closest to the query vector. Sophisticated systems use variations of the query words, synonyms, and statistical techniques. These systems rate words that appear in fewer documents statistically higher on the assumption that they are better at discriminating the search.

IR techniques are primarily at the word level. Systems based on Natural Language Processing techniques have not been able to demonstrate a significant improvement to date. When performing IR related tasks, it appears that the words convey as much information as the sentences.

Knowledge Discovery in Databases

[Frawley, 92] defines knowledge discovery as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from data." Frawley's definition assumes that data will be stored in the form of databases, but we can extend this idea to include text-based documents as well. Patterns in the data are deemed interesting when they are useful and provide insights that were previously unknown. Knowledge is useful when it can be used to solve a problem or meet an objective. Finally, the discovery process must be efficient enough to solve problems that are deemed interesting.

Data possesses inherent problems that a knowledge discovery system must consider. The first is missing or null information. If a database contains a null field for a person's middle initial, then either the person does not have a middle name or the

middle initial is missing. Which is true is impossible to ascertain. Databases can set null fields to a neutral value or prompt the user for additional information, but the problem of missing information is one that a discovery system must address.

The second problem is that of noise or uncertainty. A human user will be suspicious when a request for the average annual rainfall in Saudi Arabia is reported as 26 inches. A discovery system by its definition will seek the extremes in a data set in an attempt to deliver potentially interesting information. The problem is how to determine if the data is unusual because it is erroneous. Another aspect of this problem is uncertainty. If data is inputted on the basis of an average or some statistical measurement, then the information concerning correlation and deviation may or may not be available. The discovery system must be able to determine the quality of the information it receives.

Third, the impact of irrelevant information must be considered. Today's databases are very large and will certainly contain information that is not relevant to the current problem. The discovery system will have to make judgments on whether a potential attribute, tuple, table, or even database is relevant. This is a difficult task because discovering a solution to interesting problems may require unconventional approaches. Failing to search a particular document or database on the basis of a predetermined criteria may speed the search but could potentially ignore valuable data. Similarly, redundant information abounds in large, heterogeneous databases. [Matheus, 93] describes a common form of redundant information as "functional

dependency." This form exists when one field is defined as a function of other fields such as

Percentage_On_Hand := On_Hand / Authorized;

To prevent redundancies, KDS's must be aware of the dependencies of the databases.

Finally, the distributed, dynamic nature of present day databases pose problems for discovery systems. Current organizations add data at a far faster rate than it can be effectively analyzed, and solutions to most problems of significance depend on the latest information. Synchronization problems occur when processes are conducted in dynamic, heterogeneous environments. For example, if the task is to examine several organizations and determine which is better situated to meet a certain task, then it is important to have an "as of" time that is consistent between the organizations and have data that is as current as practicable.

Because of the problems listed above, discovery requires a significant amount of computation. In order to constrain the search process, the system uses inherent or background knowledge to focus queries to the relevant information. Background knowledge can exists in many forms, the most common being the data dictionaries of the databases. Other sources include domain knowledge or inter-field relationships such as height and weight. Domain knowledge provided by an expert eliminates search paths by providing guidelines or rules. If an organization can not satisfy its requirements without a certain piece of equipment, and the discovery system detects the unavailability of that equipment, then the search can stop. As previously discussed, however, limiting the search space can impact on the quality of the

discoveries. The system must provide a balance that minimizes search in unpromising areas, but allows search in interesting ones.

Figure B-5 depicts a potential knowledge discovery system. The system is interactive, guided by the user. The arrows illustrate the flow of data from the databases to the user through the Knowledge Discovery System and the Information Retrieval System. The Discovery System can send queries to the databases or to the Information Extraction System and can use background or previously determined knowledge from its Knowledge Base. The user can then review the results of the Discovery System and analyze the documents returned by the IR System to make a decision.

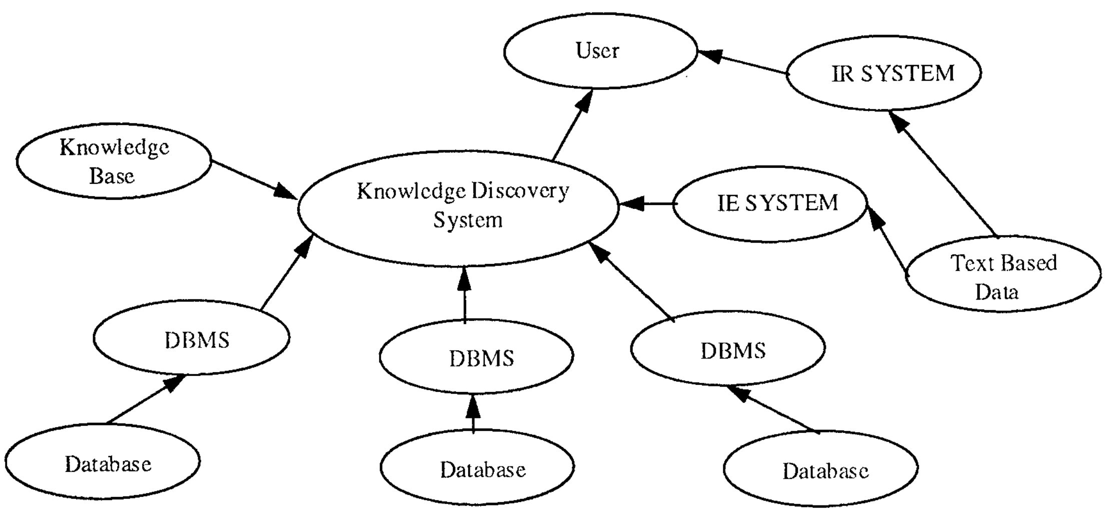


Figure B-5: Knowledge Discovery System

REFERENCES

Aha, David, Kibler, Dennis, Albert, Marc, "*Instance-Based Learning Algorithms,*" Machine Learning, Vol. 6. 1991.

Army Regulation 220-1, Unit Status Reporting for Field Organizations, Headquarters, Department of the Army, Washington DC

Army Regulation 350-1, Army Training, Headquarters, Department of the Army, Washington DC

Army Regulation 570-2, Organization and Equipment Authorization Tables, Headquarters, Department of the Army, Washington DC

Army Regulation 611-1, Personnel Selection and Classification, Headquarters, Department of the Army, Washington DC

Army Regulation 700-138, Army Logistics Readiness and Sustainability, Headquarters, Department of the Army, Washington DC

Betts, Richard K., Military Readiness, The Brookings Institution, Washington, DC, 1995.

Fisher, Doug and Lenz, Hans-J., Learning from Data, Artificial Intelligence and Statistics V, Springer-Verlag, New York, 1996.

Frawley, William J., Piatetsky-Shapiro, Matheus, Christopher, *Knowledge Discovery in Databases: An Overview*, AI Magazine, pg. 57- 65, 1992.

Gorman, Paul, and Sejnowski, Terrence, *Analysis of Hidden Units in a Layered Network Trained to Classify Sonar Targets,* Neural Networks, Vol. 1. 1988.

Haykin, Simon, Neural Networks, A Comprehensive Foundation, Macmillan Publishing, 1994.

Herz, John, Anders, Krogh, Palmer, Richard, Introduction to the Theory of Neural Computation, Addison-Wesley, Redwood City, CA, 1991.

Holz, Robert F., Hiller, Jack, and McFann, Howard, Determinants of Effective Unit Performance. US Army Research Institute for the Behavioral and Social Sciences, Alexandria, VA, July, 1994.

Norvig, Peter, Paradigms of Artificial Intelligence: Case Studies in Common Lisp, Morgan Kaufmann, San Mateo, CA, 1992.

GSA, NSIAD 95-29, "*DOD Needs to Develop a More Comprehensive Measurement System*," September, 1994.

Rumelhart, D.E., and J.L. McClelland, Parallel Distributed Processing, Vol. 1., Cambridge, MA: MIT Press.

Russell, Stuart, and Norvig, Peter, Artificial Intelligence, A Modern Approach, Prentice Hall, Upper Saddle River, New Jersey, 1995.

Schaffer, Cullen, "*Selecting a Classification Method by Cross-Validation,*" Proceedings of the Conference on Artificial Intelligence, *1992*.

Walpole, R., and Myers, R., Probability and Statistics for Engineers and Scientist, Prentice Hall, NJ, 1993.

Wilensky, Robert, Common LISPcraft, WW Norton and Company, New York, 1986.